

**UNIVERSIDADE FEDERAL DE SANTA CATARINA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA**  
**COMPUTAÇÃO**

**SILVIO COSTA SAMPAIO**

**MODELAGEM E IMPLEMENTAÇÃO**  
**ORIENTADA A OBJETOS DE UM CLIENTE DE**  
**REDE PARA BANCO DE DADOS DE IMAGENS**  
**MÉDICAS DIGITAIS UTILIZANDO O PADRÃO**  
**DICOM 3.0**

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Ciência da Computação.

Prof. Dr. rer. nat. Aldo von Wangenheim

Florianópolis, Dezembro de 1999

# **MODELAGEM E IMPLEMENTAÇÃO ORIENTADA A OBJETOS DE UM CLIENTE DE REDE PARA BANCO DE DADOS DE IMAGENS MÉDICAS DIGITAIS UTILIZANDO O PADRÃO DICOM 3.0**

Silvio Costa Sampaio

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação Área de Concentração Sistemas de Computação e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.



---

Prof. Fernando A. O. Gauthier, Dr.

Coordenador do Curso

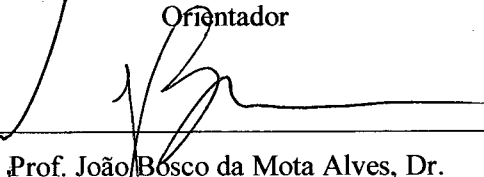
Banca Examinadora



---

Prof. Aldo von Wangenheim, Dr.

Orientador



---

Prof. João Bosco da Mota Alves, Dr.



---

Prof. Luiz Felipe Nobre. MD.



---

Prof. João Bosco Mangueira Sobral, Dr

À minha família.

## AGRADECIMENTOS

À Deus.

À minha família.

Aos colegas do Projeto Cyclops.

E a todos que, ao seu modo, contribuíram para a realização deste trabalho.

Serei sempre grato!



## SUMÁRIO

<b>LISTA DE FIGURAS .....</b>	<b>7</b>
<b>LISTA DE TABELAS.....</b>	<b>8</b>
<b>RESUMO .....</b>	<b>9</b>
<b>ABSTRACT .....</b>	<b>10</b>
<b>1 INTRODUÇÃO.....</b>	<b>11</b>
1.1 OBJETIVOS .....	11
1.1.1 <i>Objetivo específico</i> .....	11
1.2 RELEVÂNCIA.....	12
1.3 ESTADO DA ARTE.....	13
1.3.1 <i>OSIRIS</i> .....	13
1.3.2 <i>dicom3tools</i> .....	14
1.3.3 <i>CTN</i> .....	14
1.3.4 <i>OFFIS – Dicom Toolkit</i> .....	15
1.3.5 <i>dicom2</i> .....	15
1.4 ORIGINALIDADE .....	16
1.5 JUSTIFICATIVAS.....	16
<b>2 DICOM 3.0 (DIGITAL IMAGING AND COMMUNICATIONS IN MEDICINE VERSION 3.0)</b> <b>.....</b>	<b>18</b>
2.1 INTRODUÇÃO E HISTÓRICO .....	18
2.1.1 <i>Padronização</i> .....	20
2.1.1.1 Documentação do padrão DICOM 3.0 .....	21
2.1.1.2 Grupos de trabalho.....	23
2.1.1.3 Novas modalidades .....	24
2.2 MODELO DE INFORMAÇÃO E OBJETOS .....	25
2.3 CLASSES DE SERVIÇO .....	28
2.3.1 <i>Elementos de Serviço</i> .....	29
2.3.2 <i>Tipos de Classe de Serviço DICOM</i> .....	30
2.3.2.1 Teste de Conectividade: Verification Service Class.....	31
2.3.2.2 Arquivamento/Transferência de Imagens: Store Service Class (através da rede) .....	31
2.3.2.3 Media Storage Service Class .....	31
2.3.2.4 Consulta de Informações e Recuperação de Imagens: Query/Retrieve Service Class.....	32
2.3.2.5 Study Content Notification Service Class.....	32
2.3.2.6 Patient Management Service Class .....	32
2.3.2.7 Study Management Service Class.....	33
2.3.2.8 Results Management Service Class.....	33
2.3.2.9 Print Management.....	33
2.4 IMPORTÂNCIA DO DICOM PARA A RADIOLOGIA.....	33
<b>3 ESTRUTURA BÁSICA DE ARQUIVOS DICOM.....</b>	<b>35</b>
3.1 CABEÇALHO DO ARQUIVO .....	35
3.2 CONJUNTO DE DADOS DICOM.....	36
3.3 ELEMENTOS DE DADOS.....	37
3.4 ESTRUTURA DE UM ELEMENTO DE DADO COM VR EXPLÍCITO .....	39
3.5 ESTRUTURA DE UM ELEMENTO DE DADO COM VR IMPLÍCITO.....	40
3.6 GROUP LENGTH .....	40
3.7 ORDENAÇÃO DE BYTES: BIG ENDIAN X LITTLE ENDIAN.....	41
3.8 TIPOS DE ELEMENTOS DE DADOS .....	41
3.8.1 <i>Tipo 1 – elementos de dado obrigatórios</i> .....	42
3.8.2 <i>Tipo 1C – elementos de dado condicionais</i> .....	42
3.8.3 <i>Tipo 2 – elementos de dados requeridos</i> .....	42

3.8.4 Tipo 2C – elemento de dado condicional .....	43
3.8.5 Tipo 3 – elementos de dado opcionais .....	43
3.9 ELEMENTOS DE DADO DO TIPO SEQUÊNCIA.....	43
3.10 CODIFICAÇÃO DOS DADOS DE PIXEL .....	45
3.10.1 Compressão de imagens DICOM utilizando o formato JPEG.....	46
3.10.2 Compressão RLE (Run Length Encoding).....	48
<b>4 DESCRIÇÃO DO PROJETO .....</b>	<b>49</b>
4.1 O PROJETO CYCLOPS .....	49
4.1.1 Trabalhos correlatos.....	49
4.2 MODELO DE INFORMAÇÃO DA IMAGEM DICOM.....	50
4.2.1 Mapeando Objetos do Mundo Real.....	51
4.2.2 Nível Paciente.....	51
4.2.3 Nível de Estudo.....	57
4.2.4 Nível de Série.....	59
4.2.5 Nível de Imagem.....	62
4.3 A APLICAÇÃO DICOMEDITOR.....	70
4.3.2. A aplicação DICOMSeriesEditor .....	72
4.3.2. A aplicação DICOMImageEditor .....	72
<b>5 MATERIAIS E MÉTODOS .....</b>	<b>74</b>
5.1 Central Test Node (CTN) .....	74
5.2 O banco de dados miniSQL (mSQL).....	75
5.3 VisualWorks Smalltalk .....	76
<b>6 TESTES REALIZADOS.....</b>	<b>77</b>
6.1 RESULTADOS OBTIDOS.....	78
<b>7 TRABALHOS FUTUROS .....</b>	<b>79</b>
<b>CONCLUSÃO.....</b>	<b>80</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>81</b>
<b>ANEXO 1 – DIAGRAMA DE CLASSES.....</b>	<b>84</b>

## LISTA DE FIGURAS

Figura 1: O modelo DICOM 3.0 sobre camadas.....	20
Figura 2: Modelo de Entidade-Relacionamento para Classes de Serviço.....	26
Figura 3: Analogia entre construir uma sentença e sua correspondência em DICOM. ....	28
Figura 4: Estrutura básica de um arquivo DICOM.....	35
Figura 5: Estrutura de um elemento de dado com um VR explícito (VR = OB,OW,SQ ou UM).....	39
Figura 6: Estrutura de um elemento de dado com um VR explícito (VR ≠ OB,OW,SQ ou UM).....	40
Figura 7: Estrutura de um elemento de dado com um VR implícito. ....	40
Figura 8: Exemplo de um elemento de dado com VR explícito definido como uma sequência de itens (VR=SQ) de tamanho indefinido, contendo dois itens de valor explícito. ....	44
Figura 9: Exemplo de codificação de dados de pixel para uma imagem DICOM.....	46
Figura 10: Mapeando objetos do mundo real no modelo de informação DICOM. ....	51
Figura 11: Tela principal do DicomEditor.....	70
Figura 12: Configurando o servidor de imagens DICOM. ....	71
Figura 13: Tela principal do DicomSeriesEditor. ....	72
Figura 14: Tela do DicomImageEditor.....	73
Figura 15: Testes realizados. ....	77

## LISTA DE TABELAS

Tabela 1: Exemplos de Classes SOP.....	29
Tabela 2 : Valores de Sintaxe de Transferência para codificação JPEG.....	47
Tabela 3: Módulo Patient IOD.....	52
Tabela 4: Atributos do módulo Patient Relationship.....	52
Tabela 5: Atributos do módulo Patient Identification.....	53
Tabela 6: Atributos do módulo Patient Demographic.....	53
Tabela 7: Atributos do módulo Patient Medical.....	54
Tabela 8: Módulo Study IOD.....	57
Tabela 9: Módulo Series IOD.....	60
Tabela 10: Módulo Image IOD.....	63

## RESUMO

Muitos dos dispositivos radiológicos utilizados em modernas clínicas de diagnóstico por imagem codificam e trocam seus dados seguindo o protocolo padrão DICOM (*Digital Imaging Communications in Medicine*). *Cyclops*, um sistema de análise de imagens baseado em conhecimento, em desenvolvimento na Universidade Federal de Santa Catarina - Brasil, juntamente com a Universidade de Kaiserslautern - Alemanha, utiliza este protocolo para a codificação e a troca das imagens por ele analisadas. Como parte do Projeto *Cyclops*, este trabalho apresenta uma modelagem verdadeiramente orientada a objetos do padrão DICOM 3.0 e a implementação de um cliente de rede para banco de dados de imagens médicas em formato padrão DICOM 3.0 - a aplicação DicomEditor.

## ABSTRACT

*Most of the digital medical devices used in modern clinics of imaging diagnosis coding and exchange data using the DICOM (Digital Imaging Communications in Medicine) standard protocol. Cyclops, a knowledge-based image analysis system developed at the Federal University of Santa Catarina - Brazil associated with the University of Kaiserslautern - Germany, uses this protocol to coding and exchange the analysed images. In the Cyclops Project, this work presents a trully object oriented modeling of DICOM 3.0 standard and presents a network client for DICOM 3.0's medical images database - the DicomEditor application.*

## 1. INTRODUÇÃO

Neste capítulo são identificados os objetivos e apresentadas as justificativas de se desenvolver um trabalho como este. Além disso, é feita uma discussão sobre o estado da arte para o projeto em questão.

### 1.1 Objetivos

O objetivo geral deste projeto é o de desenvolver uma modelagem verdadeiramente orientada a objetos do padrão DICOM 3.0 (*Digital Imaging Communications in Medicine version 3.0*), visto que a definição das entidades de informação e serviços DICOM, apesar de bastante estruturada, carece de uma metodologia verdadeiramente orientada a objetos na sua descrição e formalização. Esta necessidade surge da própria natureza de desenvolvimento contínuo do padrão, onde novas características e/ou elementos são introduzidos ao mesmo de forma bastante dinâmica.

#### 1.1.1 Objetivo específico

Como objetivo específico, este projeto apresenta a aplicação *DicomEditor* que implementa o modelo resultante do objetivo geral. O *DicomEditor*, em linhas gerais, é um cliente de rede para banco de dados de imagens médicas digitais que utiliza o padrão DICOM 3.0 para a de/codificação, armazenamento/recuperação e transferência destas imagens.

Implementado em linguagem de programação Smalltalk, o *DicomEditor* visa validar o modelo supracitado além de fornecer uma infra-estrutura DICOM para o sistema *Cyclops*.

## 1.2 Relevância

A relevância do desenvolvimento deste projeto dentro das instituições envolvidas (UFSC e Universidade de Kaiserslautern) traz contribuições diferentes, do ponto de vista científico, tecnológico e industrial.

A relevância tecnológica da pesquisa realizada é evidente pela aplicação selecionada. A introdução de técnicas de codificação e transferência de imagens médicas digitais segundo o padrão DICOM 3.0 no sistema *Cyclops* contribui de forma significativa para um salto tecnológico qualitativo e quantitativo do sistema, melhorando aceitação do produto existente e multiplicando os benefícios alcançados pelo mesmo. O desenvolvimento do presente projeto vem suprir algumas das necessidades do sistema atual, criando uma infra-estrutura para que o mesmo possa operar em ambientes heterogêneos.

A relevância científica de realizar um trabalho de pesquisa de ponta, aplicando a formação especializada adquirida no programa de mestrado é tão importante e natural quanto fomentar a formação científica. A natureza multidisciplinar do projeto, envolvendo conceitos de várias áreas da Computação (Redes de Computadores, Engenharia de Software, Banco de Dados, Análise de Sistemas) e da Medicina (Radiologia), contribuirá para o aprimoramento desses conhecimentos, por parte das diversas instituições envolvidas no projeto *Cyclops* (clínicas, hospitais, universidades e empresas), na medida em que cada instituição que possui um domínio particular em técnicas e métodos empregados em partes deste projeto permitirá o acesso a este conhecimento através da disseminação dos resultados.

A relevância deste projeto para os parceiros universitários UFSC e Universidade de Kaiserslautern é principalmente técnico-científica, mas, por se tratar de um projeto com fins industriais, ele é também relevante para que os mesmos obtenham uma visão mais aplicada, incorporando em suas pesquisas futuras uma preocupação com a utilização prática do produto final.

A relevância industrial pode ser compreendida pela própria natureza do produto que têm consequências imediatas no ambiente de clínicas radiológicas brasileiras, como a redução dos custos de exames.



Um outro ponto importante é que o desenvolvimento deste projeto proporcionará um maior intercâmbio de conhecimento e estreitamento dos laços entre pesquisadores das diferentes instituições envolvidas e, conseqüentemente, um crescimento em pesquisa cooperativa e um fortalecimento da integração entre os pesquisadores das instituições envolvidas.

### **1.3 Estado da Arte**

Embora se saiba da existência de diversas implementações de módulos do padrão DICOM para a codificação e transmissão de imagens médicas, não há registros de implementações que contemplem numa mesma aplicação todas as características apresentadas pela aplicação desenvolvida neste projeto.

Existe atualmente um sem número de aplicações que implementam partes específicas do padrão DICOM. Entretanto, estas aplicações, em sua maioria, limitam-se à visualização e codificação/decodificação de imagens em formato padrão DICOM 3.0. Dentre estas aplicações, podemos evidenciar algumas.

#### **1.3.1 *OSIRIS***

O *OSIRIS* é uma aplicação desenvolvida pelo Hospital Universitário de Geneva que implementa um conjunto de objetos DICOM para diversas modalidades de imagem (MR, CT, US, etc.) visando basicamente sua visualização e manipulação (aplicação de filtros e manipulação de algumas características da imagem). Esta aplicação não implementa mensagens de serviços DICOM para a transmissão, armazenamento ou recuperação destas imagens, de modo que não é capaz de se comunicar com um banco de imagens médicas em conformidade com o padrão DICOM.

Apesar de ser limitada sob o ponto de vista do escopo do padrão DICOM, que vai bem mais além da simples codificação/decodificação de arquivos de imagens

médicas, estendendo-se à comunicação entre diversos elementos em conformidade com o padrão, o *OSIRIS* é hoje a mais conhecida ferramenta de manipulação de arquivos de imagens em formato padrão DICOM, sendo uma referência para qualquer implementação que o suceda. Sob este ponto de vista, a aplicação resultante deste projeto incorpora grande parte das características do *OSIRIS* e estende-se à comunicação DICOM 3.0 o que a posiciona no que há de mais atual no estado da arte.

### 1.3.2 *dicom3tools*

Outra ferramenta que é tomada como referência em se tratando de implementações DICOM é o *dicom3tools* desenvolvida pelo Dr. David A. Clunie<sup>1</sup> (considerado pela comunidade da área de informática médica como a maior referência no assunto). Esta ferramenta, desenvolvida em C++, fornece uma estrutura para implementadores do padrão DICOM 3.0.

Na verdade, o *dicom3tools* é um conjunto de bibliotecas para a manipulação de arquivos em formato DICOM 3.0, além de permitir a conversão de formatos proprietário para o formato padrão DICOM 3.0.

Como sua função é apenas servir como referência para outras implementações, o *dicom3tools* não dispõe de interfaces elaboradas, não sendo considerada uma aplicação em si.

### 1.3.3 CTN

O Central Test Node (CTN) é um conjunto de ferramentas de software, desenvolvido pela Sociedade Norte-Americana de Radiologia (RSNA), que implementam diversos aspectos do padrão DICOM 3.0.

Estas ferramentas tem como objetivo permitir que participantes do RSNA

(fabricantes de equipamentos radiológicos e grupos de pesquisa) possam testar ou demonstrar que suas implementações ou equipamentos podem se comunicar com uma implementação independente do padrão DICOM 3.0, testando com isso sua conformidade com este padrão.

O CTN, juntamente com um banco de dados relacional, implementa uma espécie de servidor de imagens DICOM. Esta ferramenta será explicada na seção Materiais e Métodos, pois foi utilizada ao longo deste trabalho.

#### 1.3.4 OFFIS – Dicom Toolkit

Esta ferramenta é conhecida como o “CTN Europeu”. Desenvolvido pela *Oldenburg University* e o centro de pesquisa CERIUM, em meio ao Comitê Europeu de Padronização, o Dicom Toolkit, conhecido como DCMTK, oferece uma biblioteca para codificação e decodificação de dados DICOM.

Com base na biblioteca DCMTK, foram desenvolvidas algumas aplicações para manipulação dos dados DICOM. Assim, temos o *DICOMscope* (um visualizador de imagens em formato DICOM, permitindo diversos tipos de processamento sobre as mesmas), o DCMIMAGE (um conjunto de ferramenta para gerenciamento de imagens DICOM), DCMPRINT (uma ferramenta para gerenciamento de impressão de imagens DICOM) e o DCMCHECK (uma ferramenta de validação e testes de imagens DICOM).

#### 1.3.5 *dicom2*

Esta ferramenta é um programa de linha de comando que permite converter imagens em formato padrão DICOM para vários outros formatos (PNG, BMP, TARGA, *raw*, etc).

---

<sup>1</sup>É também um dos responsáveis pela revisão e desenvolvimento de partes do padrão DICOM

## 1.4 Originalidade

No que diz respeito ao aspecto tecnológico, em se tratando de haver uma aplicação comercial e que representa um diferencial competitivo para os fabricantes de equipamentos de imagens médicas por trás do projeto aqui apresentado, é difícil estabelecer uma comparação completa e extensiva, em termos científicos e tecnológicos em nível internacional, sob o ponto de vista da originalidade. Entretanto, em nível nacional, este projeto é pioneiro, pois não há registros de outras tentativas nacionais. Os equipamentos de imagem médica que hoje são comercializados no Brasil e que apresentam algumas características de conformidade com o padrão DICOM 3.0 incorporam implementações estrangeiras.

São conhecidas diversas implementações para resolver problemas práticos da codificação e transferência de imagens médicas segundo o padrão DICOM 3.0, entretanto, não são muito conhecidos exemplos de sistemas operando em nível comercial que contemplem, numa mesma implementação, todas as características apresentadas pelo projeto aqui proposto.

Devido à contínua evolução do padrão DICOM, com a inclusão constante de novas modalidades e adaptações às já existentes, a forte característica de orientação a objetos apresentada neste projeto constitui-se no que de mais avançado pode-se considerar no momento para o desenvolvimento de um sistema com os requerimentos apresentados.

## 1.5 Justificativas

O desenvolvimento deste projeto justifica-se inicialmente pela necessidade do Sistema *Cyclops* por uma infra-estrutura para codificação e transferência das imagens médicas, por este analisadas, utilizando o padrão DICOM 3.0.

Além disso, este projeto justifica-se por seu caráter pioneiro, que pode ser comprovado pela não existência de um projeto similar nacional. Desta forma, o

desenvolvimento desse projeto possibilitará o domínio pelo país de uma importante tecnologia apartir da disseminação dos resultados do mesmo.

Outra justificativa para este projeto é a facilidade de expansão do projeto inicial, seja através da incorporação de novas modalidades ao padrão DICOM 3.0 ou pela atualização dos módulos iniciais da aplicação.

Uma última, e não menos importante, justificativa para o desenvolvimento deste projeto é a possibilidade de formação de recursos humanos em uma área de tecnologia de ponta e de interesse global, inicilmente no âmbito do Projeto *Cyclops* e posteriormente sendo extendido à toda a comunidade científica através da disseminação do conhecimento desenvolvido no presente projeto.

## 2. DICOM 3.0 (Digital Imaging and Communications in Medicine version 3.0)

Apesar de sua importância nas áreas de informática médica, radiologia e sistemas de informação hospitalar, o padrão DICOM não é largamente conhecido. Assim, neste capítulo, vários conceitos definidos pelo padrão são apresentados.

### 2.1 Introdução e Histórico

A fim de desenvolver um padrão que permitisse aos diversos equipamentos de imagens médicas digitais (Tomógrafo Computadorizado, Raio-X, Ecógrafo, etc) comunicarem entre si, o Colégio Americano de Radiologia (*ACR*) e a Associação Nacional dos Fabricantes de Equipamentos Elétricos (*NEMA*) formaram um comitê em 1983. O objetivo deste grupo, nomeado como *ARC-NEMA Digital Imaging and Communications Standards Committee*, era definir e desenvolver uma interface entre equipamentos de tipos diferentes. Além da especificação da conexão física, o desenvolvimento do padrão incluía ainda um dicionário dos elementos necessários na codificação e interpretação das imagens.

O comitê estudou diversos padrões propostos, mas nenhum deles satisfazia a todas as necessidades de interconexão. Foi então que a Associação Americana dos Físicos em Medicina desenvolveu um formato padrão para o armazenamento de imagens médicas digitais em meios magnéticos. Esta solução continha uma descrição da imagem e os elementos que a identificavam como uma imagem médica, como por exemplo, o nome do paciente. Esta idéia foi considerada relevante e incorporada mais tarde pelo comitê.

A primeira versão do padrão DICOM foi finalmente publicada em 1985 e a necessidade de correção dos muitos erros fez com que fosse publicada uma segunda versão do padrão em 1988. Assim, em 1988, o padrão DICOM começou a ser verdadeiramente utilizado na codificação e intercâmbio de imagens médicas digitais. Apesar de começar a ser largamente utilizado, esta segunda versão do padrão ainda não

fornecia uma comunicação confiável, em virtude das diversas modificações incorporadas à esta versão.

Na verdade, o problema foi que em 1988 (DICOM 2.0) muitos usuários queriam uma interface entre seus dispositivos de imagem e suas redes. Apesar disto já ser permitido pela versão corrente do padrão, o mesmo ainda necessitava de melhoramentos a fim de permitir uma comunicação em rede robusta. Por exemplo, um dispositivo poderia enviar uma mensagem contendo a imagem e seu cabeçalho para um outro dispositivo qualquer, mas este último não tinha como saber o que o primeiro queria fazer com as informações contidas na mensagem. Isso acontecia pois o DICOM 2.0 não foi desenvolvido exatamente para suportar comunicação em rede. Este ponto ainda era secundário para o padrão.

O ACR-NEMA decidiu que desenvolver uma interface para suportar redes requeria muito mais que adicionar partes à versão existente do padrão. Assim, resolveu pela reengenharia do processo de concepção do padrão DICOM e por adotar a orientação a objetos como novo método de concepção. Assim, o DICOM passou a adquirir uma maior modularidade.

Com isso, o desenvolvimento do padrão DICOM deu-se de forma contínua, demonstrando uma ascendente compatibilidade e confiabilidade na comunicação de imagens médicas digitais.

Adicionalmente, um rápido exame dos tipos de serviços necessários à comunicação em rede, mostrou que a definição de uma classe de serviços básicos permitiria que um processo de alto nível (na camada de aplicação) fosse capaz de se comunicar com um grande número de diferentes protocolos de rede. Assim, deu-se o desenvolvimento de uma nova versão, quase um novo projeto. Estes protocolos são modelados como uma série de camadas superpostas, também conhecidas como “pilha”. Na versão 2.0 do DICOM já existia uma “pilha” que definia uma comunicação ponto-a-ponto. Mais tarde foram inseridas, baseadas em sua popularidade e possibilidade de expansão, as pilhas *Transmission Control Protocol/Internet Protocol* (TCP/IP) e a *International Standards Organization/Open Systems Interconnection* (ISO/OSI).

A figura 1 apresenta um diagrama do modelo de comunicação desenvolvido. A filosofia básica é que uma dada aplicação de imagens médicas possa se comunicar sobre qualquer das pilhas disponíveis com qualquer outro dispositivo que utilize a mesma

pilha. Com esta nova filosofia, tornou-se possível a troca de pilhas, sempre que necessário, sem ter de reescrever todo o código dos programas de aplicação que utilizassem o padrão DICOM.

Após três anos de trabalho, com muitas sugestões da indústria e da área acadêmica, foi dado por completo o DICOM 3.0. Esta versão é muito mais abrangente e robusta que as versões anteriores.

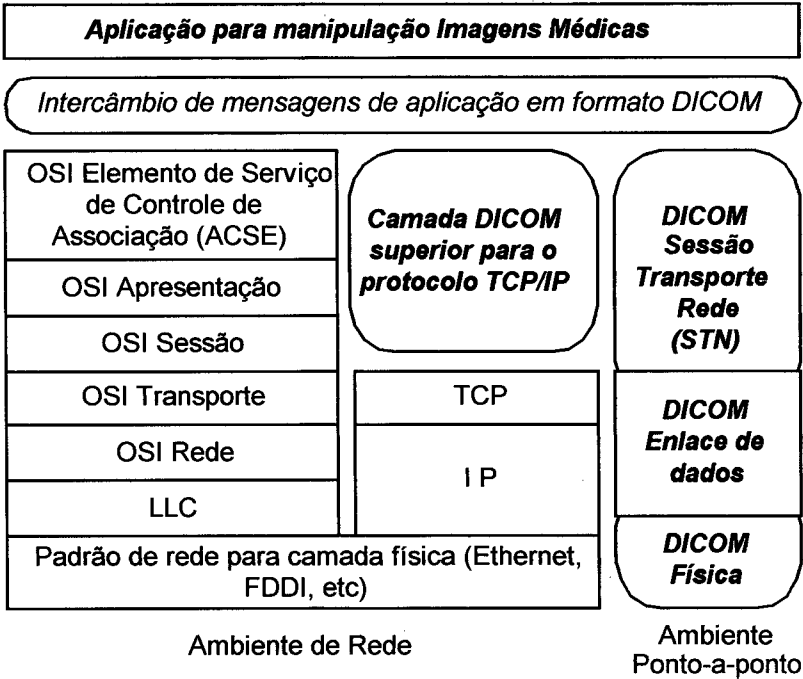


Figura 1: O modelo DICOM 3.0 sobre camadas.

Agora que o DICOM 3.0 está finalizado, usuários e fabricantes podem conhecer seu escopo e os principais conceitos envolvidos.

2.1.1 Padronização

Como visto anteriormente, o organismo responsável pela padronização e desenvolvimento do padrão DICOM é o *ARC-NEMA Digital Imaging and Communications Standards Committee*, um grupo formado em conjunto pelo Colégio



Americano de Radiologia (*ACR*) e a Associação Nacional dos Fabricantes de Equipamentos Elétricos (*NEMA*). Deste organismo participam diversos profissionais da área médica e engenheiros dos mais conhecidos fornecedores da área de equipamentos de imagem médica.

Divididos em diversos grupos este comitê desenvolveu o padrão DICOM 3.0 e produziu uma série de documentos descrevendo-o. Estes documentos são descritos a seguir.

#### **2.1.1.1 Documentação do padrão DICOM 3.0**

Diferente das versões anteriores, o padrão DICOM 3.0 divide a sua especificação em partes. Isto foi feito para que estas partes pudessem ser expandidas (por exemplo, pela adição de uma nova definição de objeto de informação - IOD) sem ter que republicar todo o padrão. Dividido em partes, onde seções de assuntos para adição ou modificação estão em anexos, reduz o trabalho de edição quando houver a necessidade de atualizações.

O padrão DICOM 3.0 está dividido em 12 partes publicadas em 1998 e designadas pelos documentos PS 3.1 até PS 3.12. A seguir temos uma breve descrição destes documentos:

**PS 3.1 *Introduction and Overview*:** uma breve introdução ao padrão DICOM. Justifica a existência do padrão e fornece uma explicação sobre a estrutura da documentação (nomeclaturas utilizadas, metodologia empregada, etc.). Para maiores informações consultar [DICOM PS3.1, 1998].

**PS 3.2 *Conformance*:** define os princípios para todas as implementações que busquem estar em conformidade com o padrão DICOM. Para maiores informações consultar [DICOM PS3.2, 1998].

**PS 3.3 *Information Object Definitions*:** descreve como os objetos de informação

são descritos. Lista todos os grupos de entidades de informação e Definições de Objetos de Informações já definidos em cada objeto de informação. Estes grupos são coleções e seqüências de elementos de dados. Para maiores informações consultar [DICOM PS3.3, 1998].

**PS 3.4 *Service Class Specifications*:** contém a especificação das Classes de Serviço. Explica como várias classes de serviço são implementadas pelas entidades de aplicação. As regras de SCU (usuário do serviço) e SCP (fornecedor do serviço) também são definidas nesta parte. É uma boa referência/tutorial para programadores de aplicações DICOM. Para maiores informações consultar [DICOM PS3.4, 1998].

**PS 3.5 *Data Structure and Encoding*:** especifica o método de codificação de dados para a transmissão e decodificação após a recepção dos dados. Para maiores informações consultar [DICOM PS3.5, 1998].

**PS 3.6 *Data Dictionary*:** provê uma lista com todas as *tags* correspondentes aos atributos dos objetos de informação que fazem parte do padrão. Para maiores informações consultar [DICOM PS3.6, 1998].

**PS 3.7 *Message Exchange*:** descreve os Serviços DICOM comumente referenciados como *DIMSE-C* e *DIMSE -N* (*DICOM Message Service Element - Composite* e *- Normalized*, respectivamente). Estes serviços são encapsulados pela comunicação DICOM. Para maiores informações consultar [DICOM PS3.7, 1998].

**PS 3.8 *Network Communication Support for Message Exchange*:** especifica o processo de negociação de associação, protocolos da camada superior, transporte dos dados e estado da máquina. Este é o componente mais complexo para implementadores DICOM. Para maiores informações consultar [DICOM PS3.8, 1998].

**PS 3.9** *Point to Point Communication Support for Message Exchange*: especifica a utilização do padrão DICOM sobre conexões ponto-a-ponto. Para maiores informações consultar [DICOM PS3.9, 1998].

**PS 3.10 - PS 3.12** *Media Storage, Application Profiles, File Format and Media Formats for Media Interchange*: explica os formatos utilizados para armazenar dados em meios removíveis como discos magneto-óticos. Para maiores informações consultar [DICOM PS3.10, 1998], [DICOM PS3.10, 1998] e [DICOM PS3.10, 1998] respectivamente.

Em meados de 1999 foram publicadas correções aos documentos acima citados. Estas correções não foram drásticas. A mudança mais significativa pode ser observada no documento PS 3.3 - *Information Object Definitions* que traz a inclusão de novos objetos de informação para a representação de novas modalidades. Ademais foram corrigidas falhas de escrita e removidas informações desnecessárias ou redundantes.

#### **2.1.1.2 Grupos de trabalho**

Atualmente o comitê *ARC-NEMA* vêm trabalhando no desenvolvimento contínuo do padrão DICOM. Para isso, está dividido em diversos grupos de trabalho, cada um com a missão de incorporar ao padrão DICOM novas características, das mais diversas áreas médicas, expandindo seu escopo.

Atualmente são em 12 os grupos de trabalho, a saber:

Grupo de Trabalho 1 - *Cardiac and Vascular Information*;

Grupo de Trabalho 2 - *Digital X-Ray*;

Grupo de Trabalho 3 - *Nuclear Medicine*;

Grupo de Trabalho 4 - *Compression*;

- Grupo de Trabalho 5 - *Exchange Media*;
- Grupo de Trabalho 6 - *Base Standard*;
- Grupo de Trabalho 7 - *Radiotherapy*;
- Grupo de Trabalho 8 - *Structured Reporting*;
- Grupo de Trabalho 9 - *Ophthalmology*;
- Grupo de Trabalho 10 - *Strategic Advisory*;
- Grupo de Trabalho 11 - *Display Function Standard*;
- Grupo de Trabalho 12 - *Ultrasound*;
- Grupo de Trabalho 13 - *Visible Light*;
- Grupo de Trabalho 14 - *Security*;
- Grupo de Trabalho 15 - *Digital Mammography*;
- Grupo de Trabalho 16 - *Magnetic Resonance*;
- Grupo de Trabalho 17 - *3D*;
- Grupo de Trabalho 18 - *Clinical Trials and Education*;
- Grupo de Trabalho 19 - *Dermatologic Standards*;
- Grupo de Trabalho 20 - *Integration of Imaging and Information Systems*;

### **2.1.1.3 Novas modalidades**

Seguindo a filosofia de desenvolvimento contínuo do padrão DICOM 3.0, em meados de 1999 foram introduzidos ao padrão DICOM 3.0 novos objetos de informação para suportar novas modalidades. Dentre elas, podemos citar as modalidades de: Patologia, Oftalmologia, *Waveform*, *Structured Reporting* e *Advanced MR*.

Dentre estas devemos uma atenção especial às modalidades de *Waveform* e *Structured Reporting*. O primeiro vêm fornecer ao padrão DICOM 3.0 a capacidade de

contemplar exames cujo resultado não se apresenta como uma imagem de pixels, mas sim como ondas, como no caso do Ecocardiograma. O segundo busca expandir o escopo do padrão atual para o gerenciamento de resultados o que acaba por facilitar a integração do RIS (*Radiology Information System*) com o HIS (*Hospital Information System*). Nos dois casos, percebemos que o desenvolvimento contínuo do padrão DICOM 3.0 tem como direção a integração do ambiente radiológico com o ambiente hospitalar.

A constante adição de novas modalidades ao padrão DICOM é o que o torna promissor como um possível padrão para a integração de ambientes radiológicos-hospitais.

## 2.2 Modelo de Informação e Objetos

O modelo de informação definido no DICOM 3.0 difere das versões anteriores em muitos aspectos. O mais importante deles é o fato de que a estrutura básica do padrão foi modificada. As versões anteriores baseavam-se em um modelo de informação implícito que é utilizado em departamentos de radiologia. Os elementos de dados foram agrupados com base apenas na experiência dos projetistas, o mapeamento foi ruim e a estrutura das mensagens permitia apenas a informação necessária a ser transmitida.

Em contraste, o DICOM 3.0 baseia-se em modelos explícitos e detalhados de como as entidades (paciente, imagem, relatório, etc.) envolvidas em operações de radiologia devem ser descritas e como elas estão relacionadas entre si. Estes modelos, chamados de Modelos de Entidade-Relacionamento (MER) e são uma maneira de se certificar que fabricantes e usuários entendam as bases do desenvolvimento das estruturas de dados utilizadas no DICOM.

A figura 2 mostra um exemplo de um MER. A vantagem deste tipo de modelo é que eles mostram claramente todos os itens de dado requeridos em um determinado cenário inicialmente modelado e como estes itens interagem e relacionam-se entre si. A importância da modelagem resulta da necessidade de se conhecer o contexto da

informação quando consideramos uma comunicação em rede.

Em ambientes ponto-a-ponto, o usuário saberá exatamente qual dispositivo está conectado a ele e quais as suas capacidades. No entanto, num ambiente multiusuário centenas de dispositivos podem ser agregados às redes e alguns deles podem ser reconfigurados dinamicamente para manipular diferentes cargas de dados ou tarefas. Isto significa que nem sempre será possível saber o que os diversos dispositivos poderão fazer. Os dispositivos então deverão negociar termos em comum antes de estabelecer uma conexão de fato a fim de construir um ambiente de comunicação necessário para executar as tarefas requisitadas pelo usuário.

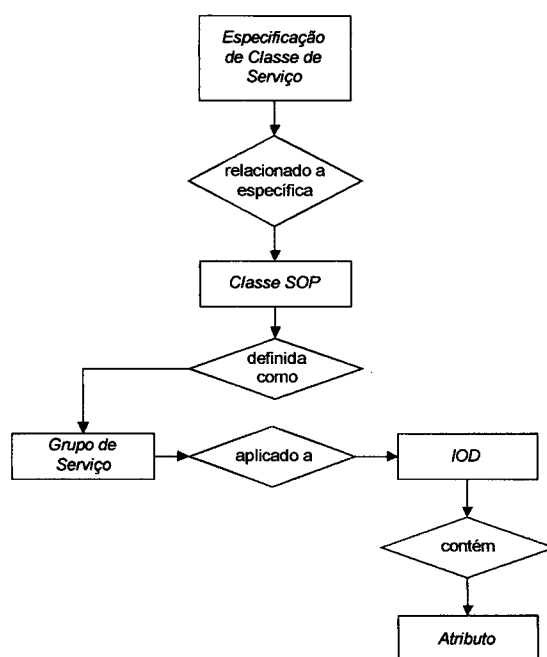


Figura 2: Modelo de Entidade-Relacionamento para Classes de Serviço.

Como exemplo, podemos considerar uma pessoa fazendo uma chamada telefônica para um estrangeiro para perguntar sobre como preparar lagostas. No nível mais baixo, a comunicação envolve discar o número de telefone correto. Isto estabelecerá um enlace entre as duas pessoas. Mas, se as duas pessoas não falam a mesma língua, eles não estarão habilitados a comunicar-se. Caso ambos falem a mesma língua, mas seus conhecimentos em receitas de lagosta sejam muito diferentes, eles não

conseguirão executar a tarefa de conversar sobre como preparar lagostas. Logo, o sucesso na comunicação requer não apenas que o primeiro indivíduo (cliente) possua o número de telefone (endereço de rede) correto do outro indivíduo (servidor) e estabeleça uma conexão telefônica, mas que eles concordem na linguagem a ser utilizada e que eles negociem um nível no qual eles possam ter um comum entendimento (contexto de apresentação).

A adoção do método de Projeto Orientado a Objetos, seguido pelo DICOM 3.0, permitiu que este pudesse ser desenvolvido utilizando-se de estruturas de dados baseadas em modelos e análise de versões de entidades abstratas do mundo real. No padrão, os objetos são as **entidades** (ou coleção de entidades) definidas através do modelo. Descrevendo as características de cada entidade, definem-se os **atributos**. Por exemplo, a entidade “paciente” possui atributos que incluem “nome do paciente” e “idade do paciente”. O padrão nomeia os objetos baseado em seus modelos de “**objetos de informação**” e os modelos e tabelas de atributos que definem estes objetos de “**definições de objetos de informação**” ou **IDOs**. As entidades representadas nos modelos são abstrações. Quando valores reais são substituídos para cada atributo, a entidade é chamada de **instância**.

O projeto orientado a objetos do DICOM também fornece uma maneira para descrever não apenas a informação, mas o que fazer com a informação ou como processos acessarão a informações sobre uma coleção de objetos. Na orientação a objetos, métodos são incorporados aos objetos, definindo seu comportamento. O padrão DICOM faz uso deste conceito para a definição de **serviços** como “armazenar uma imagem” ou “buscar informações sobre um paciente”. Estes serviços são implementados no DICOM utilizando-se construtores conhecidos como **operações** ou **notificações**. O padrão DICOM define um conjunto de operações e notificações genéricas nomeadas de **elementos de serviço de mensagens DICOM** ou **DIMSE**. A combinação de um objeto de informação e um determinado serviço é chamado de **par de serviço-objeto** ou **SOP**. Um objeto de informação pode ser usado com um conjunto de serviços, o que resulta numa **classe SOP**.

## 2.3 Classes de Serviço

Segundo o padrão DICOM, uma Classe de Serviço é um conjunto de operações (armazenar, recuperar, encontrar, mover, etc.) que podem ser executadas sobre um objeto (uma imagem de Tomografia Computadorizada, por exemplo). A definição de Classes de Serviço fornece ao DICOM uma maior modularidade, apoiada pela orientação a objetos.

Parte da Classe de Serviço é a descrição das informações e operações. Em DICOM, a descrição da informação (Objeto DICOM) e a operação a ser aplicada (Serviço) são combinadas com a definição de classes, criando o que é definido como Classe SOP (*Service Object Pair Class*). Em cada definição de Classe SOP uma única Definição de Objeto de Informação ou IOD (*Information Object Definition*) é combinada com um ou mais serviços. Para cada um destes serviços os detalhes de ambas as partes são fixados. Mais que uma Classe SOP pode existir na Classe de Serviço quando mais que uma IOD é envolvida. A figura 3 faz uma analogia entre a construção de uma sentença e a construção de uma instância SOP.

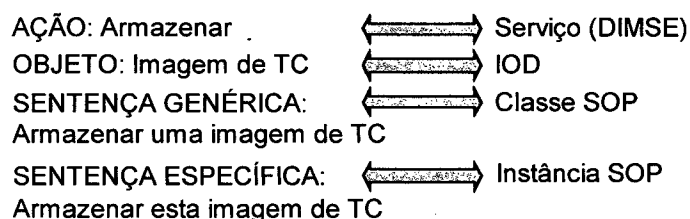


Figura 3: Analogia entre construir uma sentença e sua correspondência em DICOM.

O relacionamento entre Classes de Serviços e Classes SOP é definido pela descrição das Classes de Serviços, onde são definidas explicitamente as regras de comunicação a serem seguidas por ambos os lados. Dependendo de uma Classes de Serviço o contexto de serviço é definido. No padrão DICOM estas regras são nomeadas: Usuário de Classe de Serviço ou SCU (*Service Class User*), o cliente que requer o serviço, e Fornecedor de Classe de Serviço ou SCP (*Service Class Provider*), o servidor que executa o serviço sobre o objeto determinado.



Quando ambas as partes concordam em usar uma Classe SOP, ambas precisam garantir que estas regras sejam respeitadas exatamente como foram descritas, usando para isso o contexto encapsulado na Classe de Serviço.

Quando uma instância da Classe SOP é criada é gerada uma identificação única como atributo desta instância. A identificação é própria mais para o uso interno do sistema do que para identificação humana e possui dois aspectos: a identificação de classe e a identificação de instância.

*Tabela 1: Exemplos de Classes SOP.*

<i><b>Classe SOP</b></i>	<i><b>Modalidade</b></i>	<i><b>UID</b></i>
<i>CT Image Storage</i>	<i>CT</i>	<i>1.2.840.10008.5.1.4.1.1.2</i>
<i>MR Image Storage</i>	<i>MR</i>	<i>1.2.840.10008.5.1.4.1.1.4</i>
<i>Nuclear Medicine Image Storage</i>	<i>NM</i>	<i>1.2.840.10008.5.1.4.1.1.20</i>
<i>Ultrasound Image Storage</i>	<i>US</i>	<i>1.2.840.10008.5.1.4.1.1.6.1</i>
<i>Verification</i>	<i>-</i>	<i>1.2.840.10008.1.1</i>

Esta identificação pode ser utilizada em um ambiente multifornecedor em diferentes lugares do mundo. Para isso, um mecanismo é utilizado para gerar uma cadeia de caracteres chamados UID (*Unique Identifier*). Antes que a informação seja trocada, deve-se identificar o UID que está diretamente associado à Classe SOP, como mostrado na Tabela 1, e que é uma importante característica a qual tem que ser negociada primeiro entre as partes envolvidas na troca. O mecanismo usado depende do tipo de troca: rede ou meio físico.

### **2.3.1 Elementos de Serviço**

Elementos de Serviço são as operações permitidas sobre objetos de informação

para uma determinada Classe SOP. O grupo de Elementos de Serviço faz parte da Classe SOP e é chamado de Grupo de Serviço.

O Grupo de Serviço de uma Classe SOP é selecionado de uma lista fixa de Elementos de Serviço DICOM. Alguns Elementos de Serviço são próprios para uso com um IOD Composto e outros apenas com IOD Normalizado. A terceira categoria, Elementos de Serviço relacionados à meios de armazenamento, manipulam instâncias de Classe SOP Composta e Normalizada como arquivos.

O Contexto descrito pela Classe de Serviço é limitado quando utilizando IODs Compostos. Cada Elemento de Serviço tem um significado complexo (*STORE*, *FIND*, *MOVE*). Não há relacionamento assumido entre Elementos de Serviço individuais em uma sequência quando usamos Classe de Serviço Composto.

Em contraste, Classes de Serviço usando IODs Normalizados têm um contexto mais amplo, tal como funções de gerenciamento. Elas usam Elementos de Serviço primitivos para operações com partes simples da informação: *GET*, *SET*, *ACTION*, etc. A Classe de Serviço define a relação de uma sequência de primitivas *request*. Com Classe de Serviço Normalizada ambas as partes mantém a trilha do processamento em ambos os lados, usando Elementos de Serviço para controlá-los.

Cada Classe SOP usa um ou mais Elementos de Serviço de cada grupo composto (C - XXXX) ou grupo normalizado (N - XXXX). Os próximos Elementos de Serviço disponíveis são: *C-STORE*, *C-FIND*, *C-MOVE*, *C-GET*, *C-CANCEL*, *C-ECHO*, *N-GET*, *N-SET*, *N-ACTION*, *N-CREATE*, *N-DELETE* e *N-EVENT-REPORT*. A semântica dos Elementos de Serviço depende da Classe de Serviço e Classe SOP na qual são usados.

Elementos de Serviço relacionados a meios de armazenamento *M-WRITE*, *M-READ*, *M-DELETE*, *M-INQUIRE-FILE-SET* e *M-INQUIRE-FILE* definem funções primitivas para manipulação com conjuntos de arquivos.

### 2.3.2 Tipos de Classe de Serviço DICOM

O padrão DICOM define uma série de Classes de Serviço que podem ser

executadas sobre os objetos definidos no padrão. A seguir é apresentada a descrição dos tipos de Classes de Serviço DICOM presentes neste padrão.

#### **2.3.2.1 Teste de Conectividade: *Verification Service Class***

Utilizado apenas para fins de verificação, às vezes associado ao *DICOM Echo*, de uma conexão no nível de Aplicação entre dois Elementos de Aplicação DICOM. Sua função é similar ao comando *ping* suportado pelo sistema operacional UNIX. Enquanto *ping* verifica os níveis de *hardware* e rede de uma conexão, a Classe de Serviços de Verificação executa esta tarefa sobre uma associação DICOM.

#### **2.3.2.2 Arquivamento/Transferência de Imagens: *Store Service Class* (através da rede)**

Esta classe define uma classe de serviço do nível de aplicação a qual facilita a simples transferência de imagens. Isto permite a uma entidade de aplicação DICOM enviar imagens a uma outra. As imagens não são agrupadas em uma entidade lógica particular ou enviadas em alguma ordem definida. O dispositivo receptor (SCP) agrupa e armazena imagens de acordo com a arquitetura do banco de dados nativo. O número de UID pode ser usado para correlações lógicas das imagens.

#### **2.3.2.3 *Media Storage Service Class***

Permite o armazenamento de imagens e informações relacionadas em meios removíveis, tais como fita e CD-R. A informação armazenada pode ser lida em outro dispositivo de armazenamento ou uma impressora.

#### **2.3.2.4 Consulta de Informações e Recuperação de Imagens: *Query/Retrieve Service Class***

É uma extensão da *Storage Service Class*. Esta classe de serviço não pretende fornecer um mecanismo de consulta ao banco de dados generalizado como SQL. Em vez disso, se propõe a consultas de informações da imagem usando um pequeno conjunto de atributos chave.

Adicionalmente, a *Query/Retrieve Service Class* fornece a possibilidade para transmitir ou recuperar um conjunto de imagens identificadas. Esta capacidade permite que uma entidade de aplicação DICOM possa recuperar imagens de uma outra entidade de aplicação DICOM ou requisitar a uma entidade de aplicação DICOM remota que inicie uma transferência de imagens.

#### **2.3.2.5 Study Content Notification Service Class**

Dispositivos DICOM usam esta classe para notificar a existência, conteúdo e localização de origem de uma determinada imagem de um estudo. Utilizado para verificar se todas as imagens em um estudo estão presentes.

#### **2.3.2.6 Patient Management Service Class**

Esta classe de serviço facilita a criação e acompanhamento daquele subconjunto de informações do paciente e suas visitas que é utilizado no auxílio ao gerenciamento nos estudos radiográficos deste paciente. A principal função da *Patient Management Service Class* é suportar que entidades de aplicação requisitem acesso à informação relacionada a um determinado paciente com o propósito de obter um ou mais estudos radiográficos deste paciente.

### **2.3.2.7 Study Management Service Class**

A *Study Management Service Class* é uma classe de serviço análoga à *Patient Management Service Class*, facilitando a criação, agendamento, desempenho e acompanhamento de estudos realizados. Diferindo da maioria das classes de serviço, onde o foco é a imagem, aqui os serviços são centrados nos estudos. Um estudo é definido como uma ou mais séries de imagens. Séries, por sua vez, são um grupo de imagens que são criadas em uma modalidade (CT, NM, ultrassonografia, etc.).

### **2.3.2.8 Results Management Service Class**

Análogo a *Patient* e *Study Management Service Class*, esta classe facilita a criação e acompanhamento de resultados e interpretações associadas ao diagnóstico.

### **2.3.2.9 Print Management**

Esta classe está relacionada à impressão de imagens em dispositivos como filme ou impressora. Fornece um método flexível de formatar o *layout* do filme e manipulação de *pixels*. O usuário dá a formatação esperada e a instância desta classe faz todas as manipulações.

## **2.4 Importância do DICOM para a Radiologia**

O maior impacto do padrão DICOM certamente reside sobre os PACS (*Picture Archiving and Communications System*), permitindo o surgimento de “miniPACS” (implementações com um conjunto reduzido de funções básicas de um PACS) e

“PACS parciais” (implementações de apenas uma ou algumas das funções básicas de um PACS). Por exemplo, um sistema compartilhado de impressoras de uma unidade de imagens de CT ou MR pode utilizar o DICOM como especificação de interface de rede, interconectando os diversos equipamentos e impressoras. Num ambiente multi-fornecedor (comum em ambientes radiológicos), isto representa uma abstração das interfaces proprietárias de cada fornecedor. O efeito disto é a redução da dificuldade e custos de interconexão entre os diversos equipamentos, além de uma simplificação do trabalho (como em nosso exemplo, todos os dispositivos poderão compartilhar uma mesma impressora).

A utilidade de conectar um PACS a outro sistema de informação, particularmente um RIS (*Radiology Information System*) ou HIS (*Hospital Information System*), é extrema. Através desta conexão seria possível um total controle por parte do PACS dos exames radiológicos realizados (através do RIS) e das informações de pacientes, consultas, laudos, etc. (através do HIS). No entanto, esta conexão exige um grande trabalho por parte de analistas e programadores em definir as interfaces necessárias tanto do lado do PACS como do lado do HIS/RIS. É aí que o DICOM traz vantagens, através das classes de serviço de gerenciamento, fornecem uma maneira simplificada de resolver o problema, ao menos do lado do PACS. Como fornecedores HIS/RIS implementam seus próprios padrões, como o HL-7 (*Health Level Seven*), sua conexão com um PACS não é trivial. O grupo de pesquisas do DICOM vem se empenhando em definir interfaces padrões RIS/HIS, buscando expandir as potencialidades do padrão.

### 3. ESTRUTURA BÁSICA DE ARQUIVOS DICOM

O formato de arquivo DICOM é descrito pelo Colégio Americano de Radiologia (ACR) e a Associação Nacional dos Fabricantes de Equipamentos Elétricos (NEMA) na especificação PS3.10 “Meio de Armazenamento e Formato de Arquivos para troca de meios”, do padrão DICOM. Uma ilustração da estrutura básica de um arquivo DICOM pode ser visto a seguir:

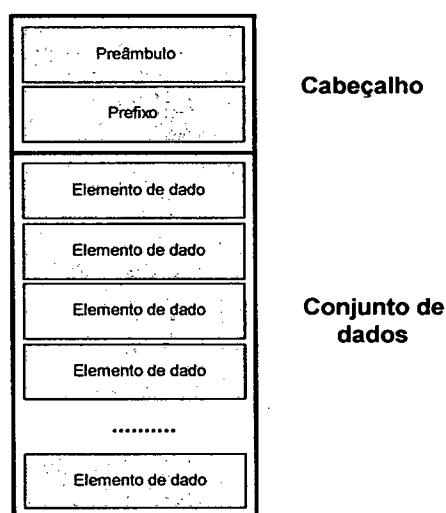


Figura 4: Estrutura básica de um arquivo DICOM.

#### 3.1 Cabeçalho do Arquivo

O cabeçalho de um arquivo DICOM possui um preâmbulo de 128 *bytes*, seguidos de um prefixo DICOM de 4 *bytes*, que pode ou não ser incluído no arquivo.

O padrão DICOM não exige nenhuma estrutura para o tamanho fixo de preâmbulo. Não é exigido que o mesmo seja estruturado como um elemento de dados DICOM com uma *Tag* e um tamanho (*Length*). Pretende-se, com isso, facilitar o acesso para as imagens e outros dados no arquivo DICOM com a condição de compatibilidade com grande parte dos formatos de arquivo de imagem usados em computadores.

Se o preâmbulo do arquivo não é usado pela aplicação ou implementação específica, todos os 128 *bytes* deverão ter valor *00H*. Isto busca facilitar o reconhecimento do preâmbulo mesmo quando este não é utilizado.

O preâmbulo do arquivo pode, por exemplo, conter informação que permita que uma aplicação multimídia qualquer possa acessar imagens arquivadas em um conjunto de dados DICOM.

Os próximos quatro *bytes* de prefixo DICOM deverão conter o *string* de caracteres “DICM” codificado como as letras maiúsculas da *ISO 8859 G0 Character Peoertoire*. Estes quatro *bytes* de prefixo não estão estruturados como um elemento de dados DICOM com uma *Tag* e um tamanho (*Length*).

### 3.2 Conjunto de dados DICOM

Cada arquivo deve conter um único conjunto de dados representando uma única instância SOP relacionada a uma única classe SOP (e correspondente IOD).

Em um arquivo contendo mais que um único *frame* de imagens 2D, *IODs* específicos podem ser definidos para incluir múltiplos *frames*.

A Sintaxe de Transferência usada para codificar o Conjunto de dados deve ser única e identificada pela *Transfer Syntax UID* da meta-informação do arquivo DICOM.

Um conjunto de dados DICOM não inclui seu comprimento total. O fim da identificação do arquivo fornecido pelo *DICOM File Sevice* é a única identificação do final do conjunto de dados.

O último elemento de dados DICOM de um conjunto de dados deve ser (FFFC, FFFC), conhecido como *Data Set Trailing Padding Data Element*. O valor deste elemento de dado não possui significado específico e deverá ser ignorado por todas as implementações DICOM quando lendo este conjunto de dados. Apesar disto, leitores de arquivos DICOM devem estar aptos a processarem este *Data Set Trailing Padding* (FFFC, FFFC) também no conjunto de dados seguindo a meta-informação ou em conjunto de dados colocados em uma sequência.

Um conjunto de dados representa uma instância do mundo real representada como um Objeto de Informação. Um conjunto de dados é composto de uma série de



elementos de dados. Elementos de dados contém o valor dos atributos codificado para este objeto. O conteúdo específico e semântico destes atributos estão especificados nas Definições do Objeto de Informação (IOD). *Pixel Data*, *Overlays* e *Curves* são elementos de dados e suas interpretações dependem de outros elementos relacionados.

### 3.3 Elementos de Dados

Um elemento de dado é constituído de três campos, são estes: *Tag*, *Value Length* e *Value*. Um quarto campo, *Value Representation*, está apenas presente em estruturas com Sintaxe de Transferência do tipo *Explicit VR Data Element*.

Um elemento de dado é unicamente identificado por uma *Tag*. Os elementos de dados em um conjunto de dados devem ser ordenados pela ordem crescente do valor da *Tag* e este deve ocorrer no máximo uma vez no conjunto de dados.

Uma *Tag* pode ocorrer outra vez apenas dentro do *Nested Data Sets*.

O campo *Tag* é um par ordenado de inteiros não-sinalizado de 16 *bits* representando o número de grupo (*Group number*) seguido pelo número do elemento dentro do grupo (*Element Number*).

*Value Representation* é um *string* de caracteres de 2 *bytes* contendo o VR do elemento de dado. O VR para um dado *Tag* deve ser definido pelo dicionário de dados como especificado no documento padrão PS3.6 do DICOM. Os dois caracteres VR devem ser codificados usando caracteres a partir do conjunto de caracteres padrão aceitos pelo DICOM.

O campo *Value Length* é um inteiro não-sinalizado de 16 ou 32 *bits* (dependendo se o VR está explícito ou implícito) que contém o comprimento explícito do valor do campo, em número de *bytes*, que representa o valor para o campo. Isto não inclui o comprimento dos campos *Tag*, *Value Representation* e *Value Length*. Este campo pode ser preenchido como um inteiro não-sinalizado de 32 *bits* com o valor (FFFFFFFFH), significando comprimento indefinido. Tamanhos indefinidos podem ser utilizados em campos com VR do tipo sequência (SQ). Pode ser também utilizado para elementos de dados com VR do tipo OW ou OB, dependendo da Sintaxe de Transferência.

O campo *Value* é um conjunto de *bytes* contendo o valor do elemento de dado. O tipo de dado deste campo é especificado pelo VR. O VR para um determinado elemento de dado pode ser determinado utilizando o Dicionário de Dados do padrão DICOM, documento OS 3.6 – *Data Dictionary*, ou consultando o campo *Value Representation* caso este esteja contido explicitamente no elemento de dado.

Outro conceito importante na definição dos elementos de dados é o valor de multiplicidade (VM). O VM especifica quantos valores, com o VR definido para este elemento, podem fazer parte do campo *Value*. Se o VM é maior que um, múltiplos valores deverão ser delimitados no campo *Value*, segundo definido na seção 6.4 do padrão DICOM. Os valores de multiplicidade também são definidos no Dicionário de Dados.

Dois tipos de elemento de dado são definidos:

1. Elemento de dado padrão (*Standard Data Element*): possui sempre um *Group Number* par que não é (0000, eeee), (0002, eeee), (0004, eeee) ou (0006, eeee). NOTA: o uso destes grupos de exceção é reservado para comandos DIMSE e formato de arquivos DICOM.
2. Elemento de dado privado (*Private Data Element*): possui um *Group number* ímpar que não é (001, eeee), (0003, eeee), (0005, eeee), (0007, eeee) ou (FFFF, eeee).

Um elemento de dado pode ser representado de duas formas, segundo o seu valor de representação (*VR*), diferindo na forma como seus tamanhos são expressados: *VR* explícito e *VR* implícito. Para ambos o elemento de dado contém o *Data Element Tag*, *Value Length* e *Value*. Elementos de dados com *VR* Explícito e Implícito não devem coexistir em um mesmo conjunto de dados. Se um conjunto de dados utiliza *VR* Explícito ou Implícito, entre outras características, isto será determinado pela negociação da Sintaxe de Transferência.

VRs não estão contidos em elementos de dados quando usando Sintaxe de Transferência padrão DICOM (*DICOM Implicit VR Little Endian Transfer Syntax*).

3.4 Estrutura de um elemento de dado com VR explícito

Quando estruturas com VR explícito são utilizadas, o elemento de dado deve ser construído com quatro campos consecutivos: *Tag*, VR, *Value Length* e *Value*. Dependendo do VR do elemento de dado, este poderá ser estruturado de três maneiras:

- 1. Para VR com valor OB, OW, SQ e UM, os 16 *bits* seguintes aos dois caracteres do campo VR serão reservados para uso futuro e deverão ser valorados como 0000H e não deverão ser utilizados ou decodificados. O campo *Value Length* deverá ser um inteiro não-sinalizado de 32 *bits*. Se o campo *Value* possuir um tamanho explícito então o campo *Value Length* deve conter o número de bytes do campo *Value*. Entretanto, se o campo *Value* possuir um tamanho indefinido, deverá ser utilizado um delimitador de sequência (*Sequence Delimitation Item*) para marcar o fim do campo *Value*.
- 2. Para VR com valor UT, os 16 *bits* seguintes aos dois caracteres do campo VR serão reservados para uso futuro e deverão ser valorados como 0000H e não deverão ser utilizados ou decodificados. O campo *Value Length* deverá ser um inteiro não-sinalizado de 32 *bits*. O campo *Value* deverá ter um tamanho explícito (VR com valor UT não pode ter tamanho indefinido).
- 3. Para todos os outros tipos de VR, o campo *Value Length* deve ser um inteiro não-sinalizado de 16 *bits* seguindo aos dois caracteres do campo VR. O campo *Value Length* deverá conter o tamanho (em bytes) do campo *Value*.

Um exemplo de um elemento de dado com um VR explícito, como seria para dados do tipo OB, OW, SQ ou UM, é mostrado na figura abaixo:

Group Number	Element Number	VR	Reservado	Value Length	Value
2 bytes	2 bytes	2 bytes	2 bytes = 0x00,0x00	4 bytes	(Value Length) bytes

Figura 5: Estrutura de um elemento de dado com um VR explícito (VR = OB,OW,SQ ou UM).

Um exemplo de um elemento de dado com um VR explícito, como seria para qualquer outro tipo de dado diferente de OB, OW, SQ ou UM, é mostrado na figura abaixo:

Group Number	Element Number	VR	Value Length	Value
2 bytes	2 bytes	2 bytes	2 bytes	(Value Length) bytes

Figura 6: Estrutura de um elemento de dado com um VR explícito (VR ≠ OB,OW,SQ ou UM)

3.5 Estrutura de um elemento de dado com VR implícito

Quando utilizando o VR implícito, o elemento de dado deve ser estruturado em três campos consecutivos: *Tag*, *Value Length* e *Value*. Se o campo *Value* tiver um valor explícito então o campo *Value Length* deverá conter um valor igual ao tamanho (em bytes) do campo *Value*. Caso o campo valor seja de tamanho indefinido, o campo *Value Length* terá o valor (FFFFFFFFH) e um delimitador de sequência (*Sequence Delimitation Item*) marcando o fim do campo *Value*.

A estrutura de um elemento de dado com um VR implícito é mostrada na figura abaixo:

Group Number	Element Number	Value Length	Value
2 bytes	2 bytes	4 bytes	(Value Length) bytes

Figura 7: Estrutura de um elemento de dado com um VR implícito.

3.6 Group Length

O *Group Length* (gggg,0000) é um elemento de dado opcional (Tipo 3) do padrão DICOM 3.0 (para maiores detalhes, consultar a seção 7.4 do padrão para uma

descrição dos tipos de elemento de dado) que fornece uma otimização opcional para esquemas de *parsing* de conjunto de dados, permitindo “pular” grupos sem interesse. Implementações podem ou não incluir este elemento de dado em um conjunto de dado DICOM. Entretanto, toda aplicação deve ser capaz de aceita-lo, mesmo que ignorando-o em seguida.

### 3.7 Ordenação de Bytes: Big Endian X Little Endian

Outro componente importante a considerar quando da codificação ou decodificação do conjunto de dados DICOM é a ordenação dos bytes. Esta ordenação pode ser feita de duas formas: Little Endian ou Big Endian.

Na forma Little Endian o byte menos significativo deve ser codificado primeiro, com os bytes restante codificados em ordem crescente de significância. Enquanto que, na forma Big Endian, o byte mais significativo deve ser codificado primeiro deve ser codificado primeiro, com os bytes restante codificados em ordem decrescente de significância.

O ordenamento dos bytes é um componente que é definido durante a Associação entre duas Entidades de Aplicação, apartir da Sintaxe de Transferência (Transfer Syntax). A Sintaxe de Transferência padrão DICOM, a qual deve ser suportada por toda Entidade de Aplicação, define o modo de ordenamento dos bytes como Little Endian.

### 3.8 Tipos de elementos de dados

Um atributo, codificado como um elemento de dado, pode ou não ser necessário em um conjunto de dados, dependendo do seu tipo.

O tipo de um atributo de um IOD ou de um atributo de uma definição de *SOP Class* é utilizado para indicar quando este atributo é obrigatório ou opcional. O tipo de um atributo também indica se o atributo é condicional (apenas obrigatório sob certas condições). Os tipos dos atributos dos IODs Compostos, são especificados no

documento PS 3.3 do padrão DICOM. Os tipos dos atributos dos IODs Normalizados, são especificados no documento padrão PS 3.4.

### **3.8.1 Tipo 1 – elementos de dado obrigatórios**

IODs e classes SOP definem o tipo 1 para elementos de dados que devem ser obrigatoriamente incluídos.

O campo *Value* deve conter dado válido, como definido pelos elementos VR e VM – especificados no documento PS 3.6 do padrão DICOM.

O campo *Value Length* não pode ser zero. Ausência de um valor válido no campo *Value*, para elementos de dado do tipo 1, é considerado uma violação do protocolo DICOM.

### **3.8.2 Tipo 1C – elementos de dado condicionais**

IODs e classes SOP definem o tipo 1 para elementos de dados que devem ser incluídos sob certas condições especificadas. Elementos do tipo 1C possuem os mesmos requisitos do tipo 1, mas sob certas condições. Há uma violação de protocolo toda vez que estas condições forem satisfeitas e o valor não for incluído. Quando as condições não forem satisfeitas o elemento não deve ser incluído no conjunto de dados.

### **3.8.3 Tipo 2 – elementos de dados requeridos**

IODs e classes SOP definem o tipo 2 para elementos de dados que devem ser obrigatoriamente incluídos. Porém, para este tipo é permitido que, no caso de não se conhecer o valor do elemento, este seja codificado com o campo *Value Length* com valor zero e sem valor para o campo *Value*. Caso o valor do elemento seja conhecido, o campo *Value* deve conter este valor, como definido pelos elementos VR e VM. Estes elementos de dado devem ser incluídos conjunto de dados e sua ausência é considerada uma violação do protocolo.

### 3.8.4 Tipo 2C – elemento de dado condicional

IODs e classes SOP definem o tipo 2C para elementos de dados que possuem alguns dos requisitos dos elementos do tipo 2, mas sob certas condições. Há uma violação de protocolo quando as condições são satisfeitas, mas o elemento não é incluído no conjunto de dados. Quando estas condições não são satisfeitas, o elemento não deve ser incluído.

### 3.8.5 Tipo 3 – elementos de dado opcionais

IODs e classes SOP definem o tipo 1 para elementos de dados que são opcionais. A ausência destes elementos no conjunto de dados não é considerada uma violação de protocolo.

Elementos de dados do tipo 3 também podem ser codificados com tamanho zero e sem valor. Entretanto, isto teria o mesmo efeito do que se o mesmo estivesse ausente do conjunto de dados.

## 3.9 Elementos de dado do tipo Sequência

O VR identificado por SQ (*Sequence*) deve ser utilizado para elementos de dado que possuam um valor que consista de zero ou mais itens, onde cada item possui um conjunto de elementos de dado. SQ fornece um esquema de codificação flexível que deve ser utilizado para estruturas simples onde há repetição de conjuntos de dados ou IODs mais complexos chamados *folders*. SQ também pode ser utilizado recursivamente para estruturas em multi-nível.

Elementos de dado com o VR do tipo SQ podem conter múltiplos itens, mas devem sempre ter um VM com valor um.

Existe três elementos de dado especiais relacionados às sequências e que não estão sujeitos às regras de VR dada pela Sintaxe de Transferência. Estes devem ser

codificados com VR implícito. São estes: *Item* (FFFE,E000), *Item Delimitation* (FFFE,E00D) e *Sequence Delimitation Item* (FFFE,E0DD).

Cada item de uma seqüência deve ser codificado como um elemento de dado padrão DICOM com a *Tag* específica de valor (FFFE,E000). A *Tag* é seguida de 4 bytes representando o campo *Length* que pode ser codificado de duas maneiras:

- 1. tamanho explícito: o número de bytes contidos no campo *Sequence Item Value* é codificado como um inteiro não-sinalizado de 32 bits. Este tamanho deve incluir o tamanho total de todos os elementos de dado que compõem este item. Este tamanho pode ter valor 00000000H caso o item não contenha dados.
- 2. tamanho indefinido: o campo *Length* possui o valor FFFFFFFFH para indicar uma seqüência de tamanho indefinido. Isto deve ser utilizado em conjunto com o elemento de dado *Sequence Delimitation Item* que deve ser incluído após o último item da seqüência.

O codificador de seqüências deverá escolher uma destas duas maneiras de codificação. Ambas deverão ser suportadas pelo decodificador de seqüências em uma aplicação.

Tag	Valor de Representação (VR)		Comprimento (Length)	Valor (Value)							
(gggg,eeee) com VR do tipo SQ	SQ	0000H (reservado)	FFFFFFFH (tamanho indefinido)	Primeiro Item			Segundo Item			delimitador (Sequence Delimitation Item)	
				Tag	Length	Value	Tag	Length	Value	Seq. Delim. Tag	Length
				(FFFE, E000)	98A5	Conjunto de dados	(FFFE, E000)	8321	Conjunto de dados	(FFFE, E0DD)	0000 0000H
4 bytes	2 bytes	2 bytes	4 bytes	4 bytes	4 bytes	98A5 bytes	4 bytes	4 bytes	8321 bytes	4 bytes	4 bytes

Figura 8: Exemplo de um elemento de dado com VR explícito definido como uma seqüência de itens (VR=SQ) de tamanho indefinido, contendo dois itens de valor explícito.



### 3.10 Codificação dos dados de pixel

O elemento de dado *Pixel Element Data* (7FE0,0010) identifica os dados de imagem, propriamente ditos (os valores de cada pixel da imagem). A estrutura deste elemento está definida no documento PS 3.3 do padrão DICOM, descrevendo a maneira como este pode ser codificado e como deve ser interpretado.

Este elemento possui o VR do tipo OW ou OB, dependendo da Sintaxe de Transferência negociada. A única diferença entre estes dois tipos de VR é que o tipo OB, um string de bytes, é indiferente ao ordenamento dos bytes (*Little Endian* ou *Big Endian*).

Dados de pixel de diferentes profundidades (*depth*) devem ser suportados. Para isso, três elementos de dado são definidos como parte da estrutura que define estes dados. São eles: *Bits Allocated* (0028,0100), *Bits Stored* (0028,0101) e *High Bit* (0028,0102).

Cada célula de pixel (conjunto de bits que compõem o pixel) deve conter um único valor de pixel. O tamanho da célula de pixel deve ser especificado pelo elemento de dado *Bits Allocated* (0028,0100). O elemento de dado *Bit Stored* (0028,0101) define a quantidade de bits, do total alocado para a célula de pixel, que é efetivamente utilizado para representar o valor do pixel. O valor deste elemento nunca deverá ser maior do que o valor de *Bits Allocated* (0028,0100). O elemento *High Bit* (0028,0102) especifica onde o bit de maior significância está armazenado. Assim, bits que não são utilizados para representar o valor de pixel podem ser utilizados para representar *overlays*, como descrito no documento PS 3.3 do padrão DICOM.

Por exemplo, em um conjunto de dados de pixel com 16 bits (2 bytes) alocados, 12 bits utilizados para armazenamento e tendo o bit 15 como bit de maior ordem, cada valor de pixel é codificado em uma palavra de 16 bits (word), com os 4 bits menos significantes de cada palavra não possuindo dado de pixel. Consulte o Anexo D da documentação padrão DICOM para outros exemplos de esquemas básicos de codificação.

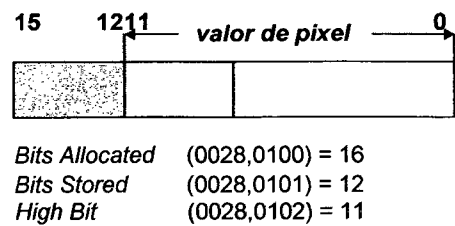


Figura 9: Exemplo de codificação de dados de pixel para uma imagem DICOM.

O campo de bits representando o valor de pixel deve ser um inteiro binário em complemento de dois ou um inteiro não-sinalizado, como especificado no elemento de dado *Pixel Representation* (0028,0103). Quando em complemento de dois, o bit de sinal deve ser o *High Bit*. O valor mínimo de pixel é especificado pelo elemento *Smallest Image Pixel Value* (0028,0106), enquanto que o valor máximo é especificado por *Largest Image Pixel Value* (0028,0107).

Dependendo da Sintaxe de Transferência negociada, em momento da associação, os dados de pixel da imagem podem estar comprimidos.

3.10.1 Compressão de imagens DICOM utilizando o formato JPEG

O padrão DICOM fornece um mecanismo para suportar a utilização de compressão de imagem JPEG, através do formato encapsulado (consultar o documento PS 3.3 do padrão DICOM). O Anexo A, da documentação do padrão, define um conjunto de Sintaxes de Transferências, como mostrado na Tabela abaixo, as quais fazem referência ao padrão JPEG e fornece um conjunto esquemas de compressão *lossless* (mínima perda de informação) e *lossy* (há perda de informação). Parte desta definição é mostrada na tabela.

Tabela 2 : Valores de Sintaxe de Transferência para codificação JPEG.

UID para Sintaxe de Transferência DICOM	Processo de codificação JPEG	Descrição JPEG
1.2.840.10008.1.2.4.50	1	baseline
1.2.840.10008.1.2.4.51	2, 4	extended
1.2.840.10008.1.2.4.52	3, 5	extended
1.2.840.10008.1.2.4.53	6, 8	spect. select. NH
1.2.840.10008.1.2.4.54	7, 9	spect. select. NH
1.2.840.10008.1.2.4.55	10, 12	full progression NH
1.2.840.10008.1.2.4.56	11, 13	full progression NH
1.2.840.10008.1.2.4.57	14	lossless NH
1.2.840.10008.1.2.4.58	15	lossless NH
1.2.840.10008.1.2.4.59	16, 18	extended H
1.2.840.10008.1.2.4.60	17, 19	extended H
1.2.840.10008.1.2.4.61	20, 22	spect. select. H
1.2.840.10008.1.2.4.62	21, 23	spect. select. H
1.2.840.10008.1.2.4.63	24, 26	full progression H
1.2.840.10008.1.2.4.64	25, 27	full progression H
1.2.840.10008.1.2.4.65	28	lossless H
1.2.840.10008.1.2.4.66	29	lossless H
1.2.840.10008.1.2.4.70	14 (Valor 1)	lossless NH, first-order prediction

O contexto de onde a compressão *lossy* de imagens médicas é clinicamente aplicável está fora do escopo do padrão DICOM. As regras associadas à seleção de parâmetros de compressão apropriados (como taxa de compressão) para a compressão JPEG *lossy* também está fora do escopo deste padrão.

De modo a facilitar a interoperabilidade de implementações, em conformidade com o padrão DICOM, que definem Sintaxes de Transferência para compressão JPEG, as seguintes regras são especificadas:

1. qualquer implementação em conformidade com o padrão DICOM que deseje suportar uma das Sintaxe de Transferência para compressão JPEG *lossless*, deve suportar o processo de compressão JPEG tipo 14 (*first-order horizontal prediction [Selection Value 1]*). O UID para esta Sintaxe de Transferência é 1.2.840.10008.1.2.4.70. Para maiores detalhes, consultar o Anexo F da documentação padrão.
2. qualquer implementação em conformidade com o padrão DICOM que deseje suportar uma das Sintaxe de Transferência para compressão JPEG *lossy* em 8-bit, deve suportar a compressão JPEG *Baseline* (processo de compressão

JPEG tipo 1). O UID para esta Sintaxe de Transferência é 1.2.840.1008.1.2.4.50.

3. qualquer implementação em conformidade com o padrão DICOM que deseje suportar uma das Sintaxe de Transferência para compressão JPEG *lossy* em 12-bit, deve suportar o processo de compressão JPEG do tipo 4. O UID para esta Sintaxe de Transferência é 1.2.840.1008.1.2.4.51.

As regras de conformidade do padrão DICOM diferenciam entre as implementações que podem apenas receber imagens codificadas em JPEG e aquelas que podem receber e processar este tipo de imagem.

O uso do formato encapsulado DICOM para suportar a compressão JPEG dos dados de pixel implica em que os elementos de dado relacionados ao formato nativo dos dados de pixel (*BitsAllocated*, *BitsStored*, *HighBit*, *PixelRepresentation* etc) devem conter valores consistentes com as características dos dados de pixel em sua forma não-comprimada. Assim, deve-se utilizar o JPEG *Interchange Format* para decodificar o *stream* JPEG e, em seguida, utilizar estes dados para interpretar os pixel da imagem.

A utilização de compressão JPEG é particularmente interessante, pois reduz, de forma substancial, o tamanho da imagem DICOM. Com isso, ganha-se em desempenho no armazenamento destas imagens, bem como em sua transmissão.

### 3.10.2 Compressão RLE (*Run Length Encoding*)

O padrão DICOM fornece ainda um mecanismo para suportar a utilização de compressão RLE, um esquema de compressão *lossless* orientado à byte aplicado através do formato encapsulado DICOM (ver documento PS 3.3 deste padrão).

O Anexo G do padrão DICOM define o uso da compressão RLE. O UID da Sintaxe de Transferência para compressão RLE é 1.2.840.10008.1.2.5.

Se o objeto codificado permitir imagens multi-frames no campo de dados de pixel, então cada frame deve ser codificado separadamente.

## 4. DESCRIÇÃO DO PROJETO

O presente projeto desenvolveu-se em duas partes principais: modelagem orientada a objetos do padrão DICOM 3.0 e implementação da aplicação *DicomEditor*.

A primeira parte gerou como resultado um modelo orientado a objetos dos objetos de informação, serviços e mensagens do padrão DICOM 3.0. A partir deste modelo abstrato, foi implementado um modelo de classes, utilizando a linguagem de programação Smalltalk, que serviu de base para a segunda parte do projeto: a implementação da aplicação *DicomEditor*.

Este capítulo descreve este processo mais detalhadamente.

### 4.1 O projeto *Cyclops*

O projeto *Cyclops*, em desenvolvimento na Universidade Federal de Santa Catarina (Brasil - coordenado pelo Prof. Dr. Aldo von Wangenheim) em cooperação com a Universidade de Kaiserslautern (Alemanha), é um laboratório para utilização e desenvolvimento de tecnologias na área de informática médica, especialmente voltado à análise automatizada de imagens médicas.

No âmbito deste projeto inúmeros trabalhos têm sido desenvolvidos visando auxiliar o diagnóstico a partir da análise de imagens médicas digitais. O trabalho aqui proposto visa oferecer ao projeto *Cyclops* uma "infra-estrutura DICOM 3.0", de modo que todos os trabalhos desenvolvidos no âmbito deste projeto possam se beneficiar das facilidades do padrão DICOM 3.0.

#### 4.1.1 Trabalhos correlatos

Como citado no item anterior, inúmeros trabalhos têm sido desenvolvidos no

âmbito do projeto *Cyclops*, onde a maior parte diz respeito à análise automatizada de imagens médicas digitais, empregando metodologias de Visão Computacional, Redes Neurais, etc. Entretanto, todos estes trabalhos relacionam-se com o projeto aqui descrito através de interfaces para a aquisição das imagens a serem analisadas (que no âmbito do projeto *Cyclops* são codificadas em formato padrão DICOM 3.0 e armazenadas num banco de dados de imagens médicas digitais) e/ou armazenamento de imagens resultantes.

Fora do escopo de análise automatizada de imagens médicas, mas que também se utiliza da infra-estrutura provida pelo presente projeto está o projeto "Sala de Laudos Virtual"<sup>2</sup>, que visa implementar uma aplicação de Teleradiologia que permita que vários profissionais médicos possam discutir e analisar casos através de uma rede local ou da Internet. Este trabalho também irá se beneficiar da estrutura proporcionada pelo presente projeto, visto que atualmente a maioria das imagens médicas digitais já são adquiridas em formato padrão DICOM, de modo que a utilização da supracitada estrutura permite que se trabalhe diretamente com estas imagens sem a necessidade de maiores transformações.

Outro trabalho de mestrado desenvolvido na UFSC, no âmbito do projeto *Cyclops*, visa transformar a aplicação *DicomEditor* num servidor de imagens médicas digitais em conformidade com o padrão DICOM 3.0.

## 4.2 Modelo de Informação da Imagem DICOM

A manipulação eletrônica de informação requer um modelo para representar a maneira como a informação deve ser estruturada. Esta estruturação é necessária para que se tenha instâncias uniformes e para que seja possível descrever as relações entre essas instâncias de forma não ambígua.

O modelo de Informação da Imagem DICOM é derivado da maneira como as imagens são manipuladas em um departamento de radiologia. As imagens são coletadas, de um ou mais dispositivos ou modalidades, e anexada a uma pasta de pacientes. Estas

---

<sup>2</sup>O trabalho "Sala de Laudos Virtual" está sendo desenvolvido no âmbito do projeto *Cyclops* pelo aluno Euclides de Moraes Barros Junior, mestrando do Curso de Ciência da Computação da Universidade

imagens são ordenadas nesta pasta com base no tipo de exame (série de imagens as quais possuem um certo relacionamento).

Os usuários de cada tipo de modalidade possuem sua própria terminologia, como por exemplo: estudo, scan, slice, etc. Quando os dados de imagem de diferentes fontes necessitam ser colecionados em um ambiente único, isso só é possível quando todos os dados de imagem estão estruturados de acordo com o mesmo modelo de informação.

#### 4.2.1 Mapeando Objetos do Mundo Real

O modelo de informação de imagem do DICOM é baseado na compreensão do modo no qual informações de diferentes modalidades são relacionadas. Os quatro níveis deste modelo de informação são: Paciente (*Patient*), Estudo (*Study*), Série (*Series*) e Imagem (*Image*).

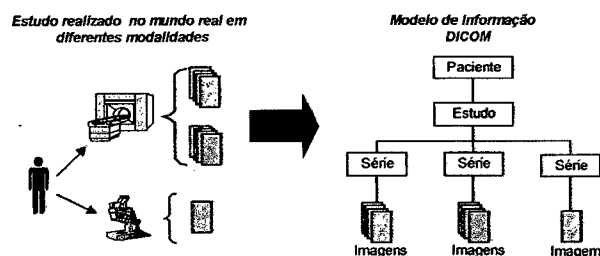


Figura 10: Mapeando objetos do mundo real no modelo de informação DICOM.

#### 4.2.2 Nível Paciente

Neste nível estão a identificação e outras informações a respeito do paciente ao qual um estudo pertence. Como mais de um estudo de um mesmo paciente pode existir, este nível é o mais alto. Entretanto, a prática normal é utilizar o nível de estudo para

coleccionar as informações manipuladas por vários sistemas sobre o determinado exame.

Na documentação do padrão DICOM o nível Paciente é descrito numa série de tabelas interligadas por referências, como mostrado abaixo.

Tabela 3: Módulo Patient IOD.

Modulo	Referência	Descrição
SOP Common	C.12.1	Contains SOP common information
Patient Relationship	C.2.1	References to related SOPs
Patient Identification	C.2.2	Identifies the real world patient
Patient Demographic	C.2.3	Describes the patient
Patient Medical	C.2.4	Medical information about patient

Fonte: Documento PS 3.3 do padrão DICOM.

A tabela 3 define o *Pacient IOD*, descrevendo os módulos que o compõem e suas referências.

Tabela 4: Atributos do módulo Patient Relationship.

Nome do atributo	Tag	Descrição do Atributo
Referenced Visit Sequence	(0008,1125)	Uniquely identifies the Visit SOP Instances associated with this Patient SOP Instance
>Referenced SOP Class UID	(0008,1150)	Uniquely identifies the referenced SOP Class.
>Referenced SOP Instance UID	(0008,1155)	Uniquely identifies the referenced SOP Instance.
Referenced Study Sequence	(0008,1110)	Uniquely identifies the Study SOP Instances associated with the Patient SOP Instance.
>Referenced SOP Class UID	(0008,1150)	Uniquely identifies the referenced SOP Class.
>Referenced SOP Instance UID	(0008,1155)	Uniquely identifies the referenced SOP Instance.
Referenced Patient Alias Sequence	(0038,0004)	Uniquely identifies any Patient SOP Instances which also describe this patient. These SOP Instances are aliases.
>Referenced SOP Class UID	(0008,1150)	Uniquely identifies the referenced SOP Class.



>Referenced SOP Instance UID	(0008,1155)	Uniquely identifies the referenced SOP Instance.
---------------------------------	-------------	--

Fonte: Documento PS 3.3 do padrão DICOM.

*Tabela 5: Atributos do módulo Patient Identification.*

Nome do Atributo	Tag	Descrição do Atributo
Patient's Name	(0010,0010)	Patient's full legal name
Patient ID	(0010,0020)	Primary hospital identification number or code for the patient
Issuer of Patient ID	(0010,0021)	Name of healthcare provider which issued the Patient ID
Other Patient IDs	(0010,1000)	Other identification numbers or codes used to identify the patient
Other Patient Names	(0010,1001)	Other names used to identify the patient
Patient's Birth Name	(0010,1005)	Patient's birth name
Patient's Mother's Birth Name	(0010,1060)	Birth name of patient's mother
Medical Record Locator	(0010,1090)	An identifier used to find the patient's existing medical record (e.g. film jacket)

Fonte: Documento PS 3.3 do padrão DICOM.

*Tabela 6: Atributos do módulo Patient Demographic.*

Nome do Atributo	Tag	Descrição do Atributo
Patient's Address	(0010,1040)	Legal address of the named patient
Region of Residence	(0010,2152)	Region within patient's country of residence
Country of Residence	(0010,2150)	Country in which patient currently resides
Patient's Telephone Numbers	(0010,2154)	Telephone numbers at which the patient can be Reached
Patient's Birth Date	(0010,0030)	Date of birth of the named patient
Patient's Birth Time	(0010,0032)	Time of birth of the named patient
Ethnic Group	(0010,2160)	Ethnic group or race of patient
Patient's Sex	(0010,0040)	Sex of the named patient. Enumerated Values: M = male F = female O = other
Patient's Size	(0010,1020)	Patient's height or length in meters

Patient's Weight	(0010,1030)	Weight of the patient in kilograms
Military Rank	(0010,1080)	Military rank of patient
Branch of Service	(0010,1081)	Branch of the military. The country allegiance may also be included (e.g. U.S. Army)
Patient's Insurance Plan Code Sequence	(0010,0050)	A sequence that conveys the patient's insurance plan.
>Code Value	(0008,0100)	The code value (defined by the coding scheme) that represents the patient's insurance plan name.
>Coding Scheme Designator	(0008,0102)	The code from table D-1 designating the coding scheme which maps the Code Value (0008,0100) onto the Code Meaning (0008,0104)
>Code Meaning	(0008,0104)	The patient's insurance plan name that is represented by the Code Value (0008,0100)
Patient's Religious Preference	(0010,21F0)	The religious preference of the patient
Patient Comments	(0010,4000)	User-defined comments about the patient

Fonte: Documento PS 3.3 do padrão DICOM.

*Tabela 7: Atributos do módulo Patient Medical.*

Attribute Name	Tag	Attribute Description
Patient State	(0038,0500)	Description of patient state (comatose, disoriented, vision impaired etc.)
Pregnancy Status	(0010,21C0)	Describes pregnancy state of patient. Enumerated Values: 0001 = not pregnant 0002 = possibly pregnant 0003 = definitely pregnant 0004 = unknown
Medical Alerts	(0010,2000)	Conditions to which medical staff should be alerted (e.g. contagious condition, drug allergies, etc.)
Contrast Allergies	(0010,2110)	Description of prior reaction to contrast agents.
Special Needs	(0038,0050)	Medical and social needs (e.g. wheelchair, oxygen, non-English-speaking etc.)
Last Menstrual Date	(0010,21D0)	Date of onset of last menstrual period

Smoking Status	(0010,21A0)	Indicates whether patient smokes. Enumerated Values: YES NO UNKNOWN
Additional Patient History	(0010,21B0)	Additional information about the patient's medical history

Fonte: Documento PS 3.3 do padrão DICOM.

Apartir destas tabelas foram identificados os atributos mais comumente utilizados e, em seguida, procedeu-se com a modelagem destes dados baseado na orientação a objetos. Como dito em seção anterior, em Smalltalk tudo é implementado como classes. No caso do nível Paciente, foram modeladas três classes distintas, representando as informações de um paciente.

Como mostrado abaixo, em código Smalltalk, a classe CyclopsPatient é definida tendo como variáveis de instância patientModule e studies. A variável patientModule deve ser instanciada apartir da classe CyclopsPatientModule, enquanto que studies deverá guardar uma referência para os estudo relacionados à este paciente.

```
Smalltalk defineClass: #CyclopsPatient
    superclass: #{Smalltalk.CyclopsDBObject}
    indexedType: #none
    private: false
    instanceVariableNames: 'patientModule studies '
    classInstanceVariableNames: "
    imports: "
    category: 'CyclopsPatientDBApp'
```

A classe CyclopsPatientModule, por sua vez, possui como variáveis de instância informações sobre o paciente instanciado: patientsName (nome do paciente), patientsID (identificador do paciente), patientsBirthDate (data de nascimento do paciente) e

patientSex (sexo do paciente).

```
Smalltalk defineClass: #CyclopsPatientModule
    superclass: #{Smalltalk.CyclopsDBObject}
    indexedType: #none
    private: false
    instanceVariableNames: 'patientsName patientID patientsBirthDate patientsSex '
    classInstanceVariableNames: "
    imports: "
    category: 'CyclopsPatientDBApp'
```

Entretanto, algumas informações sobre o paciente são temporais, mudando de estudo em estudo. Assim, como parte da modelagem foi criada a classe CyclopsPatientStudyModule, que relaciona um paciente com os valores de seus atributos temporais, com um determinado estudo. São variáveis de instância desta classe: patientsAge (idade do paciente), patientsSize (altura do paciente) e patientsWeight (peso do paciente).

```
Smalltalk defineClass: #CyclopsPatientStudyModule
    superclass: #{Smalltalk.CyclopsDBObject}
    indexedType: #none
    private: false
    instanceVariableNames: 'patientsAge patientsSize patientsWeight '
    classInstanceVariableNames: "
    imports: "
    category: 'CyclopsPatientDBApp'
```

Esta classes CyclopsPatientStudyModule será instanciada a cada estudo, como pode ser visto na seção 4.2.3. Esta modelagem do nível Paciente permite uma maior

flexibilidade no tratamento de dados relacionados ao paciente.

4.2.3 Nível de Estudo

O nível de estudo é o mais importante nível neste modelo de informação. Um estudo é o resultado de uma requisição por um determinado tipo de exame. Todas as atividades em um departamento de radiologia são centradas na correta manipulação de estudos.

Em geral, uma requisição pode envolver procedimentos de exame em diferentes modalidades. Isto resulta em uma série de uma ou mais imagens, dependendo do protocolo definido para o exame. Todos os dados de imagem são coletados juntos tendo o mesmo estudo como raiz. O mesmo paciente pode ter múltiplos estudos como resultado de outras (anteriores) requisições.

De forma análoga ao nível Paciente, o nível Estudo é descrito, na documentação padrão, por uma série de tabelas relacionadas, a seguir é mostrada apenas a tabela que define o módulo *Study IOD* e seus módulos componentes.

Tabela 8: Módulo *Study IOD*.

Módulo	Referência	Descrição do Módulo
SOP Common	C.12.1	Contains SOP common information
Study Relationship	C.4.1	References to related SOPs
Study Identification	C.4.2	Identifies the real world study
Study Classification	C.4.3	Classifies the real world study
Study Scheduling	C.4.4	Times and dates to perform study
Study Acquisition	C.4.5	Information acquired in study
Study Read	C.4.6	Reading related information

Fonte: Documento PS 3.3 do padrão DICOM.

Quando da modelagem foram criadas duas classes para representar o nível de Estudo: *CyclopsStudy* e *CyclopsGeneralStudyModule*, listadas a seguir.

```

Smalltalk defineClass: #CyclopsStudy

    superclass: #{Smalltalk.CyclopsDBObject}

    indexedType: #none

    private: false

    instanceVariableNames: 'generalStudy patientStudy series patient '

    classInstanceVariableNames: ''

    imports: ''

    category: 'CyclopsPatientDBApp'

```

A classe CyclopsStudy possui como variável de instância generalStudy (que deve ser instanciada apartir da classe CyclopsGeneralStudyModule, patientStudy (que deve ser instanciada apartir da classe CyclopsPatientStudyModule, series (coleção das das séries que fazem referência à este estudo) e patient (faz referência ao paciente do qual o estudo trata).

```

Smalltalk defineClass: #CyclopsGeneralStudyModule

    superclass: #{Smalltalk.CyclopsDBObject}

    indexedType: #none

    private: false

    instanceVariableNames: 'studyInstanceUID studyDate studyTime
referringPhysiciansName studyID studyDescription accessionNumber '

    classInstanceVariableNames: ''

    imports: ''

    category: 'CyclopsPatientDBApp'

```

A classe CyclopsGeneralStudyModule armazena informações sobre o exame propriamente dito. Tem como variáveis de instância studyInstanceUID (identificador do estudo), studyDate (data em que o estudo foi realizado), studyTime (hora em que o

estudo foi realizado), referringPhysiciansName (nome do responsável pela realização estudo), studyDescription (descrição sobre o estudo) e accessionNumber (número que identifica a aquisição).

#### 4.2.4 Nível de Série

Abaixo do nível de estudo todas as séries de imagens são colecionadas. Este nível identifica o tipo de modalidade que gerou as imagens, a data/tempo de quando a série foi feita e detalhes sobre o tipo de exame e o equipamento utilizado.

Uma série é sempre uma coleção de imagens relacionadas originárias de uma única modalidade. O modo como as imagens são agrupadas dentro de uma série depende do uso clínico. Como as imagens são adquiridas por uma modalidade não determina este agrupamento entretanto vários atributos identificarão a aquisição e poderão ser utilizados quando da visualização.

Em vários casos o relacionamento das imagens é definido pela maneira como a aquisição é feita. Quando aquisições em sequência possuem um relacionamento espacial ou temporal, as imagens resultantes podem ser agrupadas em uma série. Uma nova série deve ser criada quando o relacionamento existente entre imagens deixar de existir.

Outro critério para o agrupamento de imagens pode ser coletar imagens de uma única parte do corpo durante todo o exame da modalidade. Por exemplo, quando uma modalidade produz um conjunto de imagens do estômago de uma paciente de diferentes posições e diferentes momentos em um exame, as imagens podem ser agrupadas em uma única série.

Para cada tipo de modalidade as regras definindo o conteúdo de uma série devem ser descritos. O padrão DICOM não define o quê deve ser colecionado em um série como dependência da modalidade. As regras utilizadas por um dado sistema são partes de sua própria configuração.

Na documentação do padrão DICOM, o nível de Série é definido apartir da

seguinte tabela.

*Tabela 9: Módulo Series IOD.*

Nome do Atributo	Tag	Descrição do Atributo
Modality	(0008,0060)	Type of equipment that originally acquired the data used to create the images in this Series. See C.7.3.1.1.1 for Defined Terms.
Series Instance UID	(0020,000E)	Unique identifier of the Series.
Series Number	(0020,0011)	A number that identifies this Series.
Laterality	(0020,0060)	Laterality of (paired) body part examined. Required if the body part examined is a paired structure. Enumerated Values: R = right, L = left
Series Date	(0008,0021)	Date the Series started.
Series Time	(0008,0031)	Time the Series started.
Performing Physicians' Name	(0008,1050)	Name of the physicians administering the Series.
Protocol Name	(0018,1030)	User-defined description of the conditions under which the Series was performed.
Series Description	(0008,103E)	User provided description of the Series
Operators' Name	(0008,1070)	Technologist (s) supporting the Series.
Referenced Study Component Sequence	(0008,1111)	Uniquely identifies the Study Component SOP Instances to which the Series is related.
>Referenced SOP Class UID	(0008,1150)	Uniquely identifies the referenced SOP Class. Required if Referenced Study Component Sequence(0008,1111) is sent.
>Referenced SOP Instance UID	(0008,1155)	Uniquely identifies the referenced SOP Instance. Required if Referenced Study Component Sequence(0008,1111) is sent.
Body Part Examined	(0018,0015)	Text description of the part of the body examined. Defined Terms: SKULL, CSPINE, TSPINE, LSPINE, SSPINE, COCCYX, CHEST, CLAVICLE, BREAST, ABDOMEN, PELVIS, HIP, SHOULDER, ELBOW, KNEE, ANKLE, HAND, FOOT, EXTREMITY



Patient Position	(0018,5100)	Patient position descriptor relative to the equipment. Required for CT and MR images. See C.7.3.1.1.2. for Defined Terms and further explanation.
Smallest Pixel Value in Series	(0028,0108)	The minimum value of all images in this Series.
Largest Pixel Value in Series	(0028,0109)	The maximum value of all images in this Series.

Fonte: Documento PS 3.3 do padrão DICOM.

Na modelagem, esta tabela foi modelada em duas classes: CyclopsSeries e CyclopsGeneralSeriesModule. A primeira reúne todas as informações gerais sobre uma determinada série, integrando-a com o resto do modelo, enquanto a segunda guarda informações particulares da série.

```
Smalltalk defineClass: #CyclopsSeries
    superclass: #{Smalltalk.CyclopsDBObject}
    indexedType: #none
    private: false
    instanceVariableNames: 'generalSeries images study frameOfReference equipment
palette'
    classInstanceVariableNames: ''
    imports: ''
    category: 'CyclopsPatientDBApp'
```

A classe CyclopsSeries possui como variáveis de instância generalSeries (que deve ser instanciada apartir da classe CyclopsGeneralSeriesModule), images (coleção de imagens), study (referência ao estudo do qual a série pertence), frameOfReference, equipment (instância que identifica características do equipamento), palette (palheta de cores aplicável à série).

```

Smalltalk defineClass: #CyclopsGeneralSeriesModule

    superclass: #{Smalltalk.CyclopsDBObject}

    indexedType: #none

    private: false

    instanceVariableNames: 'modality  seriesInstanceUID  seriesNumber  seriesDate
seriesTime      protocolName      bodyPartExamined      smallestPixelValueInSeries
largestPixelValueInSeries operatorsName '

    classInstanceVariableNames: "

    imports: "

    category: 'CyclopsPatientDBApp'

```

A classe `CyclopsGeneralSeriesModule` armazena informações específicas que caracterizam a série, tendo como variáveis de instância: `modality` (identifica a modalidade que gerou a série), `seriesInstanceUID` (identificador único da série), `seriesNumber` (identificador da série dentro do estudo), `seriesDate` (data em que a série foi adquirida), `seriesTime` (hora em que a série foi adquirida), `protocolName` (nome do protocolo utilizado na aquisição), `bodyPartExamined` (parte do corpo examinada), `smallestPixelValueInSeries` (valor mínimo de pixel da série), `largestPixelValueInSeries` (valor máximo de pixel da séries), `operatorsName` (nome do responsável pela operação durante a aquisição da série).

#### 4.2.5 Nível de Imagem

O mais baixo nível do modelo de informação é o nível de imagem. Cada imagem contém informações sobre a aquisição e posicionamento bem como os dados da imagem propriamente ditos (largura, altura, bits, etc.). Dependendo do tipo de modalidade, o nível de imagem contém dados de uma imagem (single), duas imagens (biplane system) ou uma coleção de imagens (multiframe).

O uso de imagens multiframe elimina a duplicação de informação nos níveis

superiores, mas só é possível quando o relacionamento entre os quadros (frames) podem ser descritos de uma maneira simples. Por exemplo, quando o incremento em movimentos do sistema e tempo são iguais para todos os quadros.

A criação de imagens multiframe é mais complexa e consome mais recursos que a criação de uma simples imagem. O relacionamento entre frames, a capacidade da modalidade e total de dados de imagem produzido devem ser utilizados para determinar quando uma série de imagens simples ou multiframe é mais aplicável.

A tabela que define o nível de Imagem na documentação do padrão DICOM é mostrada abaixo.

Tabela 10: Módulo Image IOD.

Nome do Atributo	Tag	Descrição do Atributo
Image Number	(0020,0013)	A number that identifies this image
Patient Orientation	(0020,0020)	Patient direction of the rows and columns of the image. Required if image is part of a series which does not require the Image Plane Module. See C.7.6.1.1.1 for more explanation.
Image Date	(0008,0023)	The date the image pixel data creation started. Required if image is part of a series in which the images are temporally related.
Image Time	(0008,0033)	The time the image pixel data creation started. Required if image is part of a series in which the images are temporally related.
Image Type	(0008,0008)	Image identification characteristics. See C.7.6.1.1.2 for Defined Terms and further explanation.
Acquisition Number	(0020,0012)	A number identifying the single continuous gathering of data over a period of time which resulted in this image
Acquisition Date	(0008,0022)	The date the acquisition of data that resulted in this image started
Acquisition Time	(0008,0032)	The time the acquisition of data that resulted in this image started

Referenced Image Sequence	(0008,1140)	A sequence which provides reference to a set of Image SOP Class/Instance identifying other images significantly related to this image (e.g. post-localizer CT image) Encoded as sequence of items: (0008,1150) and (0008,1155)
>Referenced SOP Class UID	(0008,1150)	Uniquely identifies the referenced SOP Class. Required if Referenced Image Sequence(0008,1140) is sent.
>Referenced SOP Instance UID	(0008,1155)	Uniquely identifies the referenced SOP Instance. Required if Reference Image Sequence(0008,1140) is sent.
Derivation Description	(0008,2111)	A text description of how this image was derived. See C.7.6.1.1.3 for more explanation.
Source Image Sequence	(0008,2112)	A sequence which identifies the set of Image SOP Class/Instance pairs of the Images which were used to derive this Image.  Encoded as sequence of items: (0008,1150) and (0008,1155) See C.7.6.1.1.4 for further explanation.
>Referenced SOP Class UID	(0008,1150)	Uniquely identifies the referenced SOP Class. Required if Source Image Sequence(0008,2112) is sent.
>Referenced SOP Instance UID	(0008,1155)	Uniquely identifies the referenced SOP Instance. Required if Source Image Sequence(0008,2112) is sent.
Images in Acquisition	(0020,1002)	Number of images that resulted from this acquisition of data
Image Comments	(0020,4000)	User-defined comments about the image

Fonte: Documento PS 3.3 do padrão DICOM.

A fim de representar o nível de Imagem, foram modeladas, com base na tabela acima, o conjunto contendo as classes:

1. CyclopsDicomImage – representa o nível da imagem DICOM. Instancia várias outras classes contendo informações sobre aspectos diversos da imagem através dos atributos de instância: generalImage, imagePlane, imagePixel, contrastBolus, overlayPlane, voiLut e sOPCommon. Ainda possui as variáveis de instância series (faz referência à série a qual a imagem pertence), mrCachedImage (instância da classe MRCachedImage – armazena a imagem (em pixels) propriamente dita), minPixelValue e

maxPixelValue (valores mínimo e máximo de valor de pixel encontrado na imagem, respectivamente).

Smalltalk defineClass: #CyclopsDicomImage

superclass: #{Smalltalk.CyclopsDBObject}

indexedType: #objects

private: false

instanceVariableNames: 'generalImage imagePlane imagePixel contrastBolus overlayPlane volut sOPCommon series mrCachedImage writeStream minPixelValue maxPixelValue '

classInstanceVariableNames: 'palette16 palette8 '

imports: "

category: 'CyclopsPatientDBApp'

2. CyclopsGeneralImageModule – armazena informações gerais sobre a imagem. Tem como variáveis de instância: imageNumber (número da imagem na série), patientOrientation (plano em que estava o paciente durante a aquisição), imageDate (data de gravação da imagem), imageTime (hora da gravação da imagem), acquisitionNumber (número da aquisição), acquisitionTime (hora de aquisição da imagem), acquisitionTime (hora de aquisição da imagem), referencedImageSequence, sourceImageSequence e imageComments (comentários sobre a imagem).

Smalltalk defineClass: #CyclopsGeneralImageModule

superclass: #{Smalltalk.CyclopsDBObject}

indexedType: #none

private: false

instanceVariableNames: 'imageNumber patientOrientation imageDate imageTime imageType acquisitionNumber acquisitionTime acquisitionDate referencedImageSequence sourceImageSequence imageComments '

```

classInstanceVariableNames: "
imports: "
category: 'CyclopsPatientDBApp'

```

3. CyclopsImagePixelModule – armazena informações sobre a codificação dos pixels que formam a imagem. Define o modo como os pixels estão organizados no conjunto de dados DICOM. Tem como variáveis de instância `samplesPerPixel` (define a quantidade de bytes utilizada para codificar cada valor de pixel), `photometricInterpretation`, `rows` (altura da imagem), `columns` (largura da imagem), `bitsAllocated` (quantidade de bits alocados para cada célula de pixel), `bitsStored` (quantidade de bits utilizados, dentre os alocados, para codificar os valores de cada pixel), `highBit` (posição do bit mais significativo), `pixelRepresentation` (indica o método de representação utilizado para codificar o valor de pixel, onde o valor 0 significa que foi utilizado inteiros não-sinalizados e o valor 1 significa que foi utilizado a notação complemento-de-dois), `smallestImagePixelValue` e `largestImagePixelValue` (indicam o mínimo e máximo valor de pixel encontrado na imagem).

```
Smalltalk defineClass: #CyclopsImagePixelModule
```

```
    superclass: #{Smalltalk.CyclopsDBObject}
```

```
    indexedType: #none
```

```
    private: false
```

```
    instanceVariableNames: 'samplesPerPixel photometricInterpretation rows columns
bitsAllocated bitsStored highBit pixelRepresentation smallestImagePixelValue
largestImagePixelValue '
```

```
    classInstanceVariableNames: "
```

```
    imports: "
```

```
    category: 'CyclopsPatientDBApp'
```

4. **CyclopsImagePlaneModule** – armazena informações sobre o plano do paciente no momento de aquisição da imagem. Tem com variáveis de instância **pixelSpacing** (distância entre pixels adjacentes na imagem), **imageOrientationPatient** (plano em que o paciente estava no momento de aquisição da imagem), **imagePositionPatient** (posição do paciente no momento de aquisição), **sliceThickness** (espessura dos cortes), **sliceLocation** (posicao do primeiro pixel do corte com base em uma dada referência).

```
Smalltalk defineClass: #CyclopsImagePlaneModule
```

```
    superclass: #{Smalltalk.CyclopsDBObject}
```

```
    indexedType: #none
```

```
    private: false
```

```
    instanceVariableNames: 'pixelSpacing imageOrientationPatient imagePositionPatient
sliceThickness sliceLocation '
```

```
    classInstanceVariableNames: "
```

```
    imports: "
```

```
    category: 'CyclopsPatientDBApp'
```

5. **CyclopsContrastBolusModule** – armazena informações sobre a administração do contraste utilizado durante a aquisição da imagem. Utilizados em imagens provenientes de exames com uso de contraste. Tem como variáveis de instância **contrastBolusAgent** (identifica o tipo de contraste utilizado), **contrastBolusRoute** (caminho através do qual o contraste foi aplicado), **contrastBolusVolume** (volume de contraste utilizado), **contrastBolusStartTime** e **contrastBolusStopTime** (hora de início e fim da administração de contraste) e **contrastBolusTotalDose** (dose total de contraste utilizada).

```
Smalltalk defineClass: #CyclopsContrastBolusModule
```

```
    superclass: #{Smalltalk.CyclopsDBObject}
```

```

indexedType: #none

private: false

instanceVariableNames: 'contrastBolusAgent contrastBolusRoute contrastBolusVolume
contrastBolusStartTime contrastBolusStopTime contrastBolusTotalDose '

classInstanceVariableNames: "

imports: "

category: 'CyclopsPatientDBApp'

```

6. **CyclopsOverlayPlaneModule** – armazena informações sobre objetos do tipo *overlay* que possam, eventualmente, acompanhar a imagem. Esta classe representa um dos meios de se codificar este tipo de objeto, a outra forma é utilizando-se dos bits dos dados de pixel que não estejam sendo utilizados para armazenar valores de pixel.

```

Smalltalk defineClass: #CyclopsOverlayPlaneModule

    superclass: #{Smalltalk.CyclopsDBObject}

    indexedType: #none

    private: false

    instanceVariableNames: 'overlayRows overlayColumns overlayType overlayOrigin
overlayBitsAllocated overlayBitPosition overlayData '

    classInstanceVariableNames: "

    imports: "

    category: 'CyclopsPatientDBApp'

```

7. **CyclopsVOILUTModule** – armazena informações que definem características importantes à visualização da imagem. Dentre suas variáveis de instância, as mais utilizadas são `windowCenter` e `windowWidth`, que juntas definem a “janela” a ser aplicada à imagem, ou seja, a faixa de valores de cor de interesse.



Smalltalk defineClass: #CyclopsVOILUTModule

superclass: #{Smalltalk.CyclopsDBObject}

indexedType: #none

private: false

instanceVariableNames: 'voilutSequence lutDescriptor lutExplanation lutData  
windowCenter windowWidth windowCenterAndWidthExplanation '

classInstanceVariableNames: "

imports: "

category: 'CyclopsPatientDBApp'

8. CyclopsSOPCommonModule – armazena informações sobre o a classe SOP envolvendo o objeto Imagem. Tem como variáveis de instância sOPClassUID (identificador único da classe SOP), sOPInstanceUID (identificador único da instância SOP), instanceCreationDate (data da criação da instância SOP), instanceCreationTime (hora da criação da instância SOP) e instanceCreationUID (identificador único da criação da instância SOP).

Smalltalk defineClass: #CyclopsSOPCommonModule

superclass: #{Smalltalk.CyclopsDBObject}

indexedType: #none

private: false

instanceVariableNames: 'sOPClassUID sOPInstanceUID instanceCreationDate  
instanceCreationTime instanceCreatorUID '

classInstanceVariableNames: "

imports: "

category: 'CyclopsPatientDBApp'

Todos os dados destas classes são necessários para representar o nível de Imagem DICOM. Após instanciadas, todas estas informações ficam disponíveis na instância da classe CyclopsDicomImage.

### 4.3 A aplicação DICOMEditor

A aplicação DicomEditor, mostrada na figura 4, é o resultado da implementação do modelo orientado a objetos gerado na primeira fase do projeto. A intenção inicial era apenas desenvolver uma metodologia de validação para o modelo, entretanto, com os constantes melhoramentos e a adição de funções, esta aplicação tomou uma conotação industrial, podendo hoje ser condiderado um produto em fase de melhorias.

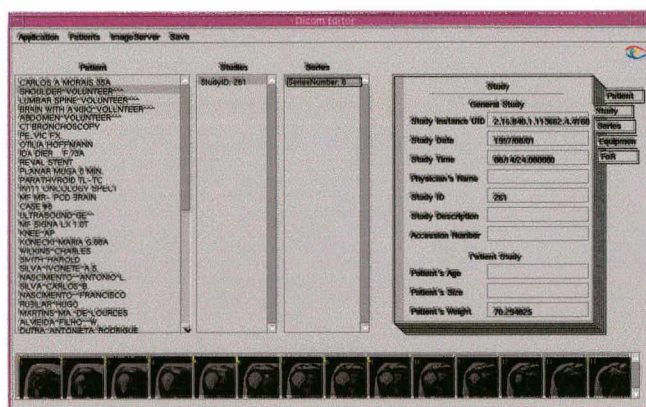


Figura 11: Tela principal do DicomEditor.

Basicamente, a aplicação DicomEditor visa ser um cliente de rede capaz de se comunicar com um banco de dados de imagens médicas em formato DICOM 3.0 e operar estas imagens internamente ou servi-las à aplicações superiores, servindo neste caso de interface.

Seguindo o modelo orientado a objetos, a aplicação DicomEditor implementa um modelo de classes utilizando a linguagem de programação Smalltalk. Estas classes representam os objetos de informação e os elementos e mensagens de serviços DICOM.

Com o DicomEditor é possível estabelecer uma comunicação com qualquer banco de dados de imagens médicas em conformidade com o padrão DICOM 3.0. Para isso, basta que sejam informados o nome do *host* que esteja servindo como servidor do banco de dados de imagem e o número de porta<sup>3</sup> para a comunicação. Estes passos São mostrados na figura 12.

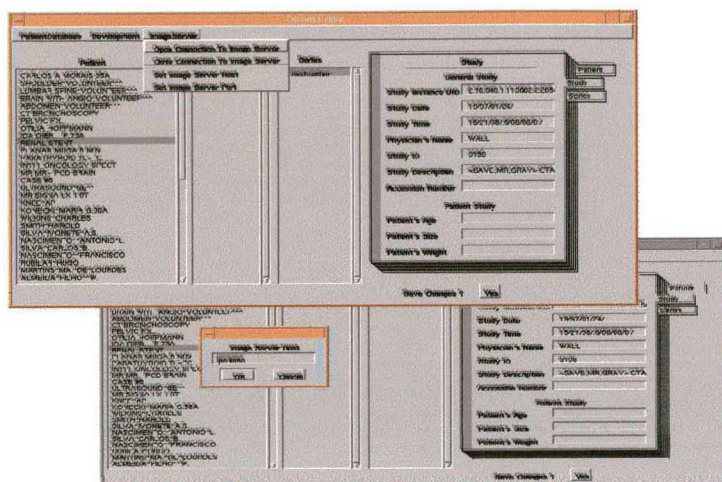


Figura 12: Configurando o servidor de imagens DICOM.

Tendo configurado o DicomEditor com as informações sobre o servidor de imagens, é possível conecta-los e executar as operações de recuperação e armazenamento de imagens. Para isso, o primeiro passo é carregar a lista de pacientes do servidor, em seguida carregar os estudos para o paciente de interesse e depois carregar as series para o estudo de interesse. Uma vez executados estes passos, a aplicação apresenta no "bloco de notas" (que fica do lado direito da tela da aplicação) informações sobre o paciente, o estudo e a série selecionados.

Apartir deste ponto, pode-se carregar todas as imagens contidas na série. Em seguida pode-se utilizar uma sub-aplicação para a visualização das imagens contidas na série. Esta sub-aplicação é chamada de DicomSeriesEditor.

<sup>3</sup>Porta, neste contexto, quer dizer um número que identifica a interface através da qual dois programas irão se comunicar. Como estamos tratando de comunicação TCP/IP, este valor de porta representa o *socket* que será estabelecido.



### 4.3.2. A aplicação DICOMSeriesEditor

A aplicação DicomSeriesEditor, mostrada na figura 13, permite que sejam visualizadas todas as imagens de uma série de imagens selecionada. Esta aplicação apresenta de forma amigável todo o conteúdo da série. Apartir desta ferramenta é possível ainda extrair informações da imagem (como valores de um pixel e de sua vizinhança, por exemplo).



Figura 13: Tela principal do DicomSeriesEditor.

### 4.3.2. A aplicação DICOMImageEditor

Interface utilizada para visualizar uma imagem previamente selecionada no DicomSeries Editor. Esta interface disponibiliza informações sobre o paciente e a própria imagem. Possui algumas ferramentas para trabalhar com a imagem, como: zoom, inserir texto em uma determinada região da imagem, um medidor de distancia (em milímetros), um campo para editar o laudo do paciente, opção para enviar a imagem em análise para outros radiologistas via e-mail, e outras utilidades que médicos radiologistas podem utilizam para estudo da imagem.

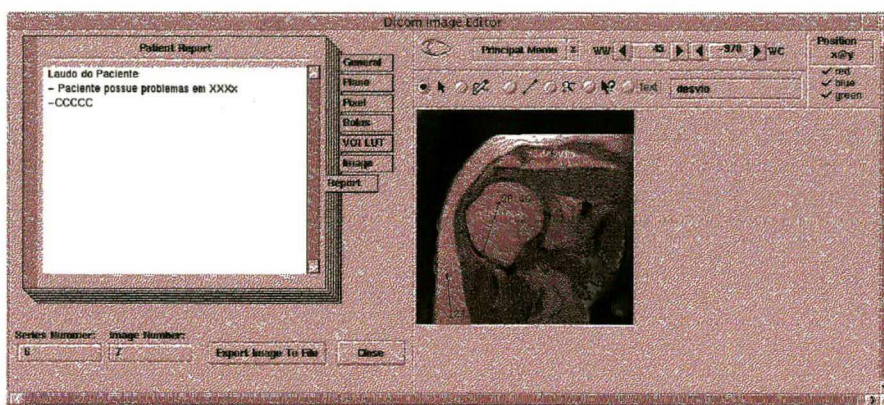


Figura 14: Tela do DicomImageEditor.

## 5. MATERIAIS E MÉTODOS

De modo a permitir que fossem testadas as características de cliente de rede para banco de dados de imagens médicas digitais da aplicação DicomEditor, fez-se necessário montar uma estrutura de Servidor de Imagens Médicas em conformidade com o padrão DICOM 3.0. Para isso, foi utilizada a aplicação CTN (*Central Test Node*) desenvolvida pela Sociedade Americana de Radiologia (RSNA) e que é capaz de se comunicar com dispositivos em conformidade com o padrão DICOM 3.0 e criar um modelo relacional para que as informações relacionadas às imagens possam ser armazenadas num banco de dados relacional comum.

Infelizmente, a aplicação CTN não suporta banco de dados robustos (Oracle, por exemplo). Assim, utilizou-se o banco de dados Mini-SQL (mSQL) como banco de dados relacional.

### 5.1 Central Test Node (CTN)

O Central Test Node (CTN) é um conjunto de ferramentas de software desenvolvido pelo RSNA que implementam aspectos do padrão DICOM 3.0. Estas ferramentas tem como objetivo permitir que participantes do RSNA (fabricantes de equipamentos radiológicos e grupos de pesquisa) possam testar ou demonstrar que suas implementações ou equipamentos podem se comunicar com uma implementação independente do padrão DICOM 3.0, testando com isso sua conformidade com este padrão.

O CTN foi implementado, de fato, pelo Electronic Radiology Laboratory do Mallinckrodt Institute of Radiology (MIR) e em 1997 foi disponibilizado aos participantes do RSNA para que estes pudessem efetuar seus testes antes da reuniões do grupo. Mais tarde o CTN foi disponibilizado em diversos servidores de ftp, tornando o mesmo de domínio público.



Dentro das capacidades oferecidas pelo CTN, está a implementação de partes das classes de serviço query/retrieve e storage do padrão DICOM 3.0, incluindo os serviços C-FIND, C-MOVE (através dos quais é capaz de listar o conteúdo armazenado na base de dados, como dados de pacientes, lista de estudos, imagens etc.) e C-STORE (utilizado para enviar imagens para o banco de dados).

O CTN utiliza a rede para comunicar-se com outros dispositivos em conformidade com o padrão DICOM 3.0. Para isto, o mesmo utiliza a pilha TCP/IP.

A utilização do CTN no presente projeto justifica-se pela necessidade de aferir as características da aplicação desenvolvida, principalmente no que diz respeito à conectividade e conformidade com o padrão DICOM 3.0.

## **5.2 O banco de dados miniSQL (mSQL)**

O banco de dados Mini SQL, ou simplesmente mSQL, é uma modesta implementação de base de dados relacional que fornece rápido acesso à dados armazenados com pouca exigência de recursos. Como o próprio nome sugere, o mSQL implementa apenas um subconjunto da linguagem SQL voltado à interfaces de consulta, não implementando maiores recursos, como views, sub-consultas, etc.

O mSQL oferece um significativo subconjunto das características fornecidas pelo SQL ANSI. Isso permite a um programa ou usuário armazenar, manipular e recuperar dados em uma tabela de estruturas. Entretanto, não são suportadas operações relacionais mais elaboradas, como união de múltiplas tabelas, visões etc.

O pacote mSQL, que pode ser adquirido gratuitamente na URL: <http://www.hughes.com.au>, inclui o banco de dados, um programa monitor, um programa que permite administrar o banco de dados, um programa visualizados de esquema e uma API (Application Program Interface) escrita na linguagem C. Como dito anteriormente, a decisão de utilizar o mSQL apoia-se no fato de que o CTN, em nenhuma de suas versões, suporta um banco de dados relacional de maior porte, além disso, o mSQL é gratuito.

### 5.3 VisualWorks Smalltalk

Smalltalk uma linguagem orientada a objetos. Por muitos, considerada a linguagem mais coerente com a orientação a objetos, esta linguagem implementa conceitos sólidos de objetos, classes, herança, polimorfismo etc. O VisualWorks® é um ambiente de programação comercial que implementa as especificações da linguagem Smalltalk.

A linguagem Smalltalk é interessante por permitir um rápido desenvolvimento, o que a torna adequada à prototipação, método que foi amplamente utilizado ao longo deste trabalho. Por esta característica, o Smalltalk é largamente utilizado em projetos de pesquisa.

Em Smalltalk, tudo é implementado como uma classe. Isso facilitou adequar a modelagem, realizada em parte deste projeto, gerando uma ferramenta de testes.



## 6. TESTES REALIZADOS

A fim de validar a implementação aqui descrita, foram realizados testes no sentido de aferir as características aqui mencionadas.

Para testar a capacidade da aplicação gerada de decodificar imagens em formato DICOM, foram utilizados diversos tipos de arquivos em formato padrão DICOM, de diversas fontes, para leitura. A aplicação se mostrou eficiente para grande parte das imagens testadas. Dentre as limitações foram identificadas: imagens comprimidas em formato JPEG, imagens *multi-frame*.

Para testar a capacidade de codificação de imagens em formato padrão DICOM, foram exportadas pela aplicação várias imagens. Em seguida, utilizou-se diversos visualizadores DICOM, disponíveis na Internet, para verificar se a codificação ocorreu com sucesso. Este teste também apresentou resultados satisfatórios.

Para testar a comunicação com o Servidor de Imagens foram realizados testes sobre as Classes de Serviço *Query/Retrieve* do padrão DICOM 3.0, executando ações de C-STORE (utilizado para armazenar um objeto de informação), C-FIND (utilizado para localizar um objeto de informação) e C-GET (utilizado para recuperar um objeto de informação) como mostrado na figura 9.

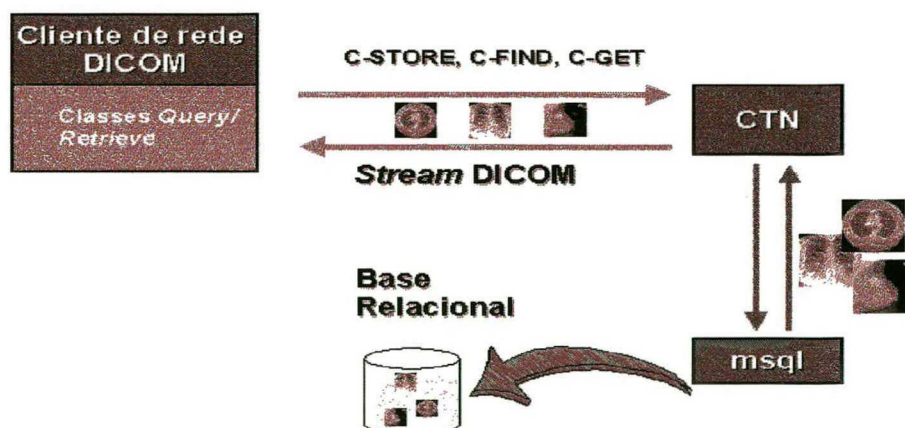


Figura 15: Testes realizados.

## **6.1 Resultados obtidos**

Os resultados dos testes com a execução ações de C-STORE, C-FIND e C-GET sobre o CTN demonstraram a conformidade da modelagem com o padrão DICOM 3.0, uma vez que obtivemos sucesso na execução destas ações.

A comunicação entre o DicomEditor e o servidor de imagens, através de mensagens padrão DICOM 3,0, também foi bem sucedida.

O modelo se mostrou exato também para a codificação e decodificação de imagens em formato DICOM 3.0, onde exaustivos testes foram realizados.

## 7. TRABALHOS FUTUROS

Apartir deste trabalho, uma série de trabalhos podem vir a ser realizados. Por ser pioneiro em nível nacional, espera-se que a disseminação do conhecimento aqui adquirido encorage novas implementações. Além disso, trabalhos de melhoramentos podem ser realizados nos resultados iniciais deste projeto.

Como possibilidades de extensão para a aplicação DicomEditor, podemos identificar algumas possibilidades:

- Implementação desta modelagem utilizando-se de outras metodologias, ferramentas ou plataformas;
- Integração do DicomEditor com RIS/HIS através da implementação de partes do padrão DICOM 3.0 que tratam do gerenciamento de fluxo de informações;
- Implementação do serviço de impressão DICOM;
- E muitas outras que podem surgir apartir do conhecimento proporcionado pelo presente projeto.

## CONCLUSÃO

A modelagem aqui apresentada mostrou-se robusta e em conformidade com o padrão DICOM 3.0. Isto foi verificado através da implementação e testes da aplicação DicomEditor.

As potencialidades da utilização do padrão DICOM no ambiente de clínicas e hospitais podem ser expandidas com a possibilidade de se mesclar tecnologias diferentes. Um bom exemplo seria a utilização da tecnologia CORBA, juntamente com o padrão DICOM, para a teleradiologia.

A aplicação DicomEditor, por sua vez, superou as expectativas do projeto se mostrando uma ferramenta útil e de interesse industrial. Em breve esta aplicação estará sendo introduzida no ambiente de clínicas parceiras do projeto *Cyclops*.

O padrão DICOM 3.0 mostrou-se bastante robusto e abrangente, sendo uma promessa de integração e interoperabilidade no ambiente radiológico-hospitalar.

Os conhecimentos de ponta adquiridos com a realização do presente projeto, certamente, representam sua maior contribuição científica. Os objetivos geral e específico foram atingidos.

## REFERÊNCIAS BIBLIOGRÁFICAS

CLUNIE, David A.. *David Clunie's Medical Image Format Site*. [online] Disponível na Internet via WWW.URL: <http://idt.net/~dclunie/index.html>.

DICOM Group. *DICOM Group Homepage*. [online] Disponível na Internet via WWW.URL: [http:// dicom.rad.unipi.it/](http://dicom.rad.unipi.it/).

*DICOM: introduction and freeware*. [online] Disponível na Internet via WWW.URL: <http://www.psychology.nottingham.ac.uk/staff/cr1/dicom.html>.

*DICOM: The standard that works*. [online] Disponível na Internet via WWW.URL: <http://dicom.rad.unipi.it/works/>.

*dicom2, a free medical image and DICOM converter*. [online] Disponível na Internet via WWW.URL: <http://www.hds.utc.fr/~barre/medical/dicom2/>.

GREVERA, George J. et al. *A WWW to DICOM interface*. SPIE Proceedings. Vol 3035, 1998.

GREVERA, George J. et al. *Implementing desktop image access of GI images*. SPIE Proceedings. Vol 3035, 1998. Vol 2435, 1996.

*Interactive DICOM Image Transmission and Tele-diagnosis over the European ATM Network*. [online] Disponível na Internet via WWW.URL: <http://dicom.rad.unipi.it/IEEE/>.

KITH. *DICOM: Digital Image and Communications in Medicine*. [online] Disponível na Internet via WWW.URL: <http://www.innherred-sykehus.no/fg4/dicom.htm>



***Medical Images transmission on heterogeneous networks.*** [online] Disponível na Internet via WWW.URL: <http://dicom.rad.unipi.it/CRS4/>.

MERGE TECHNOLOGIES INC. ***The Official DICOM Resources Page: Presented by the Committee for the Advancement of DICOM.*** [online] Disponível na Internet via WWW.URL: <http://www.merge.com/DICOM/>.

NATIONAL ELECTRICAL MANUFACTURERS ASSOCIATION. ***Digital Imaging and Communications in Medicine (DICOM);. Part 1: Introduction and Overview,*** 1998.

\_\_\_\_\_. \_\_\_\_\_. ***; Part 3: Information Model Virginia,*** 1998.

\_\_\_\_\_. \_\_\_\_\_. ***; Part 4: Service Class Especifications. Virginia,*** 1998.

\_\_\_\_\_. \_\_\_\_\_. ***; Part 5: Data Structures and Encoding. Virginia,*** 1998.

\_\_\_\_\_. \_\_\_\_\_. ***; Part 6: Data Dictionary. Virginia,*** 1998.

\_\_\_\_\_. \_\_\_\_\_. ***; Part 7: Message Exchange. Virginia,*** 1998.

\_\_\_\_\_. \_\_\_\_\_. ***; Part 8: Network Communications Support for Message Exchange. Virginia,*** 1998.

\_\_\_\_\_. \_\_\_\_\_ ; *Part 9: Point to Point Communication Support for Message Exchange*. Virginia, 1998.

\_\_\_\_\_. \_\_\_\_\_ ; *Part 10: Media Storage*. Virginia, 1998.

\_\_\_\_\_. \_\_\_\_\_ ; *Part 11: Application Profiles*. Virginia, 1998.

\_\_\_\_\_. \_\_\_\_\_ ; *Part 12: File Format and Media Formats for Interchange*. Virginia, 1998.

NEMA. ***Official Homepage of DICOM***. [online] Disponível na Internet via WWW.URL: <http://www.nema.org/nema/medical/dicom/>.

OFFIS. [online] Disponível na Internet via WWW.URL: <http://www.offis.uni-oldenburg.de/projekte/dicom/>.

RSNA. ***DICOM: The Value and Importance of an Imaging Standard***. [online] Disponível na Internet via WWW.URL: <http://www.rsna.org/REG/practiceres/dicom/index.html>.

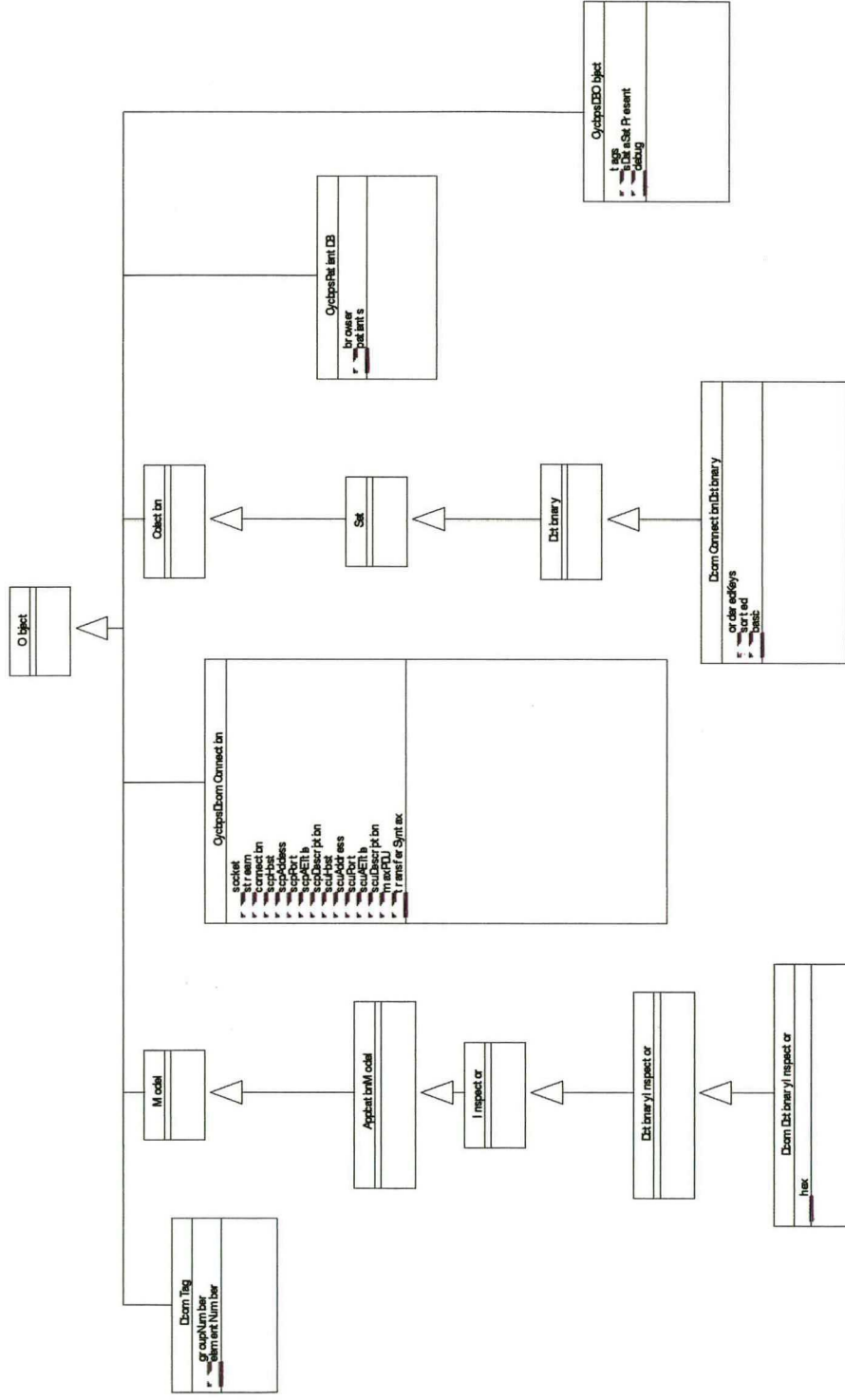
RSNA. ***RSNA Link***. [online] Disponível na Internet via WWW.URL: <http://www.rsna.org/REG/index.html>.

***Unité d'Imagerie Numérique: OSIRIS and Papyrus***. [online] Disponível na Internet via WWW.URL: <http://www.expasy.ch/www/UIN/UIN.html>.

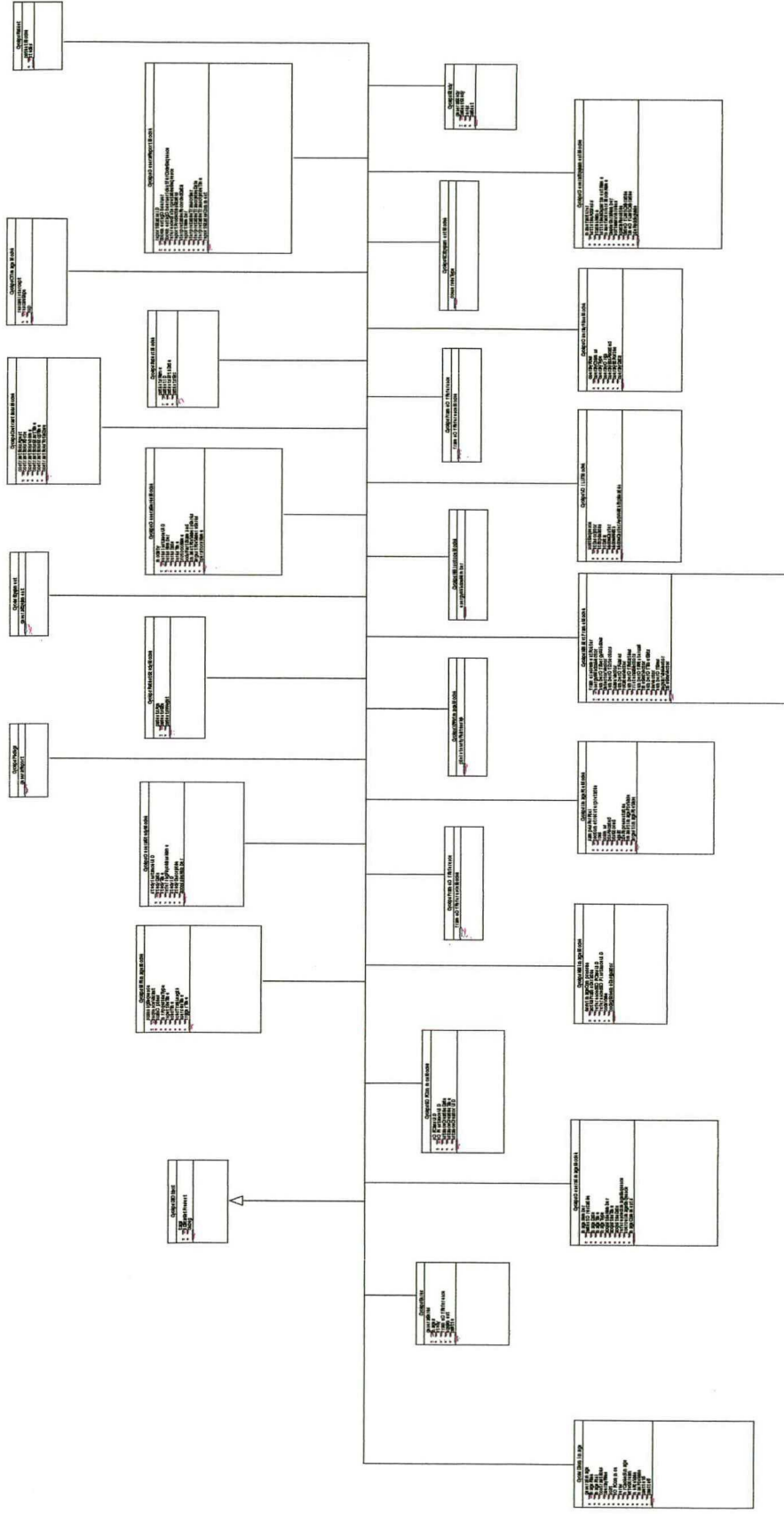




# ANEXO 1 – Parte 1/3



# ANEXO 1 – Parte 2/3



## ANEXO 1 – Parte 3/3

